TMC-3D

Motion Controller and Drive

TECHNOSOFT

Motion Controllers

Technical Reference

© Technosoft 2008

TECHNOSOFT

TMC-3D

Technical Reference

P091.048.TMC3D.UM.0708

Technosoft S.A.

Buchaux 38

CH-2022 Bevaix, NE

Switzerland

Tel.: +41 (0) 32 732 5500

Fax: +41 (0) 32 732 5504

contact@technosoftmotion.com

www.technosoftmotion.com

Read This First

Whilst Technosoft believes that the information and guidance given in this manual is correct, all parties must rely upon their own skill and judgment when making use of it. Technosoft does not assume any liability to anyone for any loss or damage caused by any error or omission in the work, whether such error or omission is the result of negligence or any other cause. Any and all such liability is disclaimed.

All rights reserved. No part or parts of this document may be reproduced or transmitted in any form or by any means, electrical or mechanical including photocopying, recording or by any information-retrieval system without permission in writing from Technosoft S.A.

The information in this document is subject to change without notice.

About This Manual

This book is a technical reference manual for the **TMC-3D**, a multi-axis motion controller with embeded digital drive. In order to operate the TMC-3D, you need to pass through the following steps:

- **Step 1 Hardware installation**
- **Step 2 Motion controller setup** using Technosoft **EasyMotion Studio** software for motion controller commissioning
- **Step 3 Slave setup** (if any)
- **Step 4 Motion programming** using one of the options:
  - The **multi-axis motion controller** executing a Technosoft Motion Language (**TML**) program developed using Technosoft **EasyMotion Studio** software
  - A **TML_LIB motion library for PCs** (Windows or Linux)
  - A **distributed control** approach which combines the above options, like for example the motion controller calling motion functions programmed on the drives in TML

This manual covers **Step 1** in detail. It describes the TMC-3D hardware including the technical data, connectors and wiring diagrams needed for installation. The manual also presents an overview of the following steps, and includes the scaling factors between the real SI units and the drive internal units. For detailed information regarding the next steps, refer to the related documentation.

Notational Conventions

This document uses the following conventions:

TML – Technosoft Motion Language

SI units – International standard units **(**meter for length, seconds for time, etc.)

IU units – Internal units of the drive

Related Documentation

Help of the EasyMotion Studio software – describes how to use the EasyMotion Studio to create motion programs using in Technosoft Motion Language (TML). EasyMotion Studio platform includes **EasySetUp** for the motion controller/drive setup, and a **Motion Wizard** for the motion programming. The Motion Wizard provides a simple, graphical way of creating motion programs and automatically generates all the TML instructions. With EasyMotion Studio you can fully benefit from a key advantage of Technosoft drives – their capability to store and execute complex motions applications without requiring an external host device. **A demo version of EasyMotion Studio (with EasySetUp part fully functional) can be downloaded free of charge from Technosoft web page.**

TML_LIB v2.0 **(part no. P091.040.v20.UM.xxxx)** – explains how to program in **C, C++,C#, Visual Basic or Delphi Pascal** a motion application for the Technosoft motion controllers/intelligent drives using TML_LIB v2.0 motion control library for PCs. The TML_lib includes ready-to-run examples that can be executed on **Windows** or **Linux** (x86 and x64).

TML_LIB_LabVIEW v2.0 **(part no. P091.040.LABVIEW.v20.UM.xxxx)** – explains how to program in **LabVIEW** a motion application for the Technosoft motion controllers/intelligent drives using TML_LIB_Labview v2.0 motion control library for PCs. The TML_Lib_LabVIEW includes over 40 ready-to-run examples.

If you Need Assistance …

If you want to …	**Contact Technosoft at …**
Visit Technosoft online	World Wide Web: http://www.technosoftmotion.com/
Receive general information or assistance (see Note)	World Wide Web: http://www.technosoftmotion.com/ Email: contact@technosoftmotion.com
Ask questions about product operation or report suspected problems	Fax: (41) 32 732 55 04 Email: hotline@technosoftmotion.com
Make suggestions about, or report errors in documentation.	Mail: Technosoft SA Buchaux 38 CH-2022 Bevaix, NE Switzerland

Contents

Read This First I

1. Safety information 1

1.1. Warnings 1
1.2. Cautions 2

2. Product Overview 3

2.1. Introduction 3
2.2. Key Features 4
2.3. Supported Motor-Sensor Configurations 5
2.4. TMC 3D Dimensions 11
2.5. Electrical Specifications 11

3. Step 1. Hardware Installation 19

3.1. Mounting 19
3.2. Connectors and Connection Diagrams 20
3.2.1. Connectors Layout 20
3.2.2. Identification Labels 21
3.2.3. Motor & Supply – J2 Connector 22
3.2.4. Feedback – J13 Connector (TMC 3D) 32
3.2.5. Analog & Digital I/O – J9 Connector 38
3.2.6. Serial Communication – J4 Connector 43
3.2.7. CAN Communication – J10 Connector 44
3.2.8. Connectors Type and Mating Connectors 47
3.3. DIP-Switch Settings 47
3.4. LED Indicators 49
3.5. First Power-Up 49

4. Step 2. Motion Controller Setup 50
4.1. Installing EasyMotion Studio 50
4.2. Getting Started with EasyMotion Studio 50
4.3. Establish communication 51
4.4. Motion Controller Application 52
4.5. Embeded Drive Setup 53
4.6. Changing the Axis ID of the Motion Controller 54
4.7. Setting CANbus Rate 55
4.8. Download Setup Data to EEPROM 56
4.9. Evaluate the behaviour of motion controller drive part (optional) 57
4.10. Creating an Image File with the Setup Data 57

5. Step 3. Slave Axes Setup 57

6. Step 4. Motion Programming 60
6.1. Multi-axis configuration 60
6.2. Generate coordinated multi-axis profiles 62
6.2.1. Coordinated multi axis modes and application management 62
6.2.2. Creating an Image File with the Setup Data and the TML Program 64
6.3. Advanced features 64
6.3.1. Error handling 64
6.3.2. Using TML Functions to Split Motion between Motion Controller and slaves axes 65
6.3.3. Customizing the Homing Procedures 65
6.3.4. Customizing the Drive Reaction to Fault Conditions 66
6.4. Using Motion Libraries for PC-based Systems 66

7. Scaling Factors 67
7.1. Position units 67
7.1.1. Brushless / DC brushed motor with quadrature encoder on motor 67
7.1.2. Brushless motor with absolute SSI encoder on motor 68
7.1.3. DC brushed motor with quadrature encoder on load and tacho on motor ... 68
7.1.4. DC brushed motor with absolute SSI encoder on load and tacho on motor 68
7.1.5. Stepper motor open-loop control. No feedback device 69

7.1.6. Stepper motor closed-loop control. Incremental encoder on motor 69
7.1.7. Stepper motor open-loop control. Incremental encoder on load 69
7.2. Speed units 70
7.2.1. Brushless / DC brushed motor with quadrature encoder on motor 70
7.2.2. Brushless motor with absolute SSI encoder on motor 70
7.2.3. DC brushed motor with quadrature encoder on load and tacho on motor ... 71
7.2.4. DC brushed motor with absolute SSI encoder on load and tacho on motor 71
7.2.5. DC brushed motor with tacho on motor 71
7.2.6. Stepper motor open-loop control. No feedback device 72
7.2.7. Stepper motor open-loop control. Incremental encoder on load 72
7.2.8. Stepper motor closed-loop control. Incremental encoder on motor 72
7.3. Acceleration units 73
7.3.1. Brushless / DC brushed motor with quadrature encoder on motor 73
7.3.2. Brushless motor with absolute SSI encoder on motor 73
7.3.3. DC brushed motor with quadrature encoder on load and tacho on motor ... 74
7.3.4. DC brushed motor with absolute SSI encoder on load and tacho on motor 74
7.3.5. DC brushed motor with tacho on motor 75
7.3.6. Stepper motor open-loop control. No feedback device 75
7.3.7. Stepper motor open-loop control. Incremental encoder on load 76
7.3.8. Stepper motor closed-loop control. Incremental encoder on motor 76
7.4. Jerk units 77
7.4.1. Brushless / DC brushed motor with quadrature encoder on motor 77
7.4.2. Brushless motor with absolute SSI encoder on motor 77
7.4.3. DC brushed motor with quadrature encoder on load and tacho on motor ... 78
7.4.4. DC brushed motor with absolute SSI encoder on load and tacho on motor 78
7.4.5. Stepper motor open-loop control. No feedback device 78
7.4.6. Stepper motor open-loop control. Incremental encoder on load 79
7.4.7. Stepper motor closed-loop control. Incremental encoder on motor 79
7.5. Current units 79
7.6. Voltage command units 79
7.7. Voltage measurement units 80
7.8. Time units 80
7.9. Drive temperature units 81
7.10. Motor position units 81
7.10.1. Brushless / DC brushed motor with quadrature encoder on motor 81

7.10.2. Brushless motor with absolute SSI encoder on motor 81
7.10.3. DC brushed motor with quadrature encoder on load and tacho on motor 82
7.10.4. DC brushed motor with absolute SSI encoder on load and tacho on motor 82
7.10.5. Stepper motor open-loop control. No feedback device 82
7.10.6. Stepper motor open-loop control. Incremental encoder on load 82
7.10.7. Stepper motor closed-loop control. Incremental encoder on motor 82
7.11. Motor speed units 83
7.11.1. Brushless / DC brushed motor with quadrature encoder on motor 83
7.11.2. Brushless motor with absolute SSI encoder on motor 83
7.11.3. DC brushed motor with quadrature encoder on load and tacho on motor 84
7.11.4. DC brushed motor with absolute SSI encoder on load & tacho on motor 84
7.11.5. DC brushed motor with tacho on motor 84
7.11.6. Stepper motor open-loop control. No feedback device or incremental encoder on load 85
7.11.7. Stepper motor closed-loop control. Incremental encoder on motor 85

8. Memory Map 87

1. Safety information

Read carefully the information presented in this chapter before carrying out the drive installation and setup! It is imperative to implement the safety instructions listed hereunder.

This information is intended to protect you, the drive and the accompanying equipment during the product operation. Incorrect handling of the drive can lead to personal injury or material damage.

Only qualified personnel may install, setup, operate and maintain the drive. A “qualified person” has the knowledge and authorization to perform tasks such as transporting, assembling, installing, commissioning and operating drives.

The following safety symbols are used in this manual:

WARNING!	***SIGNALS A DANGER TO THE OPERATOR WHICH MIGHT CAUSE BODILY INJURY. MAY INCLUDE INSTRUCTIONS TO PREVENT THIS SITUATION***

CAUTION!	***SIGNALS A DANGER FOR THE DRIVE WHICH MIGHT DAMAGE THE PRODUCT OR OTHER EQUIPMENT. MAY INCLUDE INSTRUCTIONS TO AVOID THIS SITUATION***

CAUTION!	***INDICATES AREAS SENSITIVE TO ELECTROSTATIC DISCHARGES (ESD) WHICH REQUIRE HANDLING IN AN ESD PROTECTED ENVIRONMENT***

1.1. Warnings

WARNING!	***THE VOLTAGE USED IN THE DRIVE MIGHT CAUSE ELECTRICAL SHOCKS. DO NOT TOUCH LIVE PARTS WHILE THE POWER SUPPLIES ARE ON***

WARNING!	***TO AVOID ELECTRIC ARCING AND HAZARDS, NEVER CONNECT / DISCONNECT WIRES FROM THE DRIVE WHILE THE POWER SUPPLIES ARE ON***

WARNING!	***THE DRIVE MAY HAVE HOT SURFACES DURING OPERATION.***

WARNING!	***DURING DRIVE OPERATION, THE CONTROLLED MOTOR WILL MOVE. KEEP AWAY FROM ALL MOVING PARTS TO AVOID INJURY***

1.2. Cautions

CAUTION!	***THE POWER SUPPLIES CONNECTED TO THE DRIVE MUST COMPLY WITH THE PARAMETERS SPECIFIED IN THIS DOCUMENT***

CAUTION!	***TROUBLESHOOTING AND SERVICING ARE PERMITTED ONLY FOR PERSONNEL AUTHORISED BY TECHNOSOFT***

CAUTION!	***THE DRIVE CONTAINS ELECTROSTATICALLY SENSITIVE COMPONENTS WHICH MAY BE DAMAGED BY INCORRECT HANDLING. THEREFORE THE DRIVE SHALL BE REMOVED FROM ITS ORIGINAL PACKAGE ONLY IN AN ESD PROTECTED ENVIRONMENT***

To prevent electrostatic damage, avoid contact with insulating materials, such as synthetic fabrics or plastic surfaces. In order to discharge static electricity build-up, place the drive on a grounded conductive surface and also ground yourself.

2. Product Overview

2.1. Introduction

The **TMC 3D** multi-axis motion controller incorporates in the same unit a multi-axis motion controller and a fully digital intelligent servo drive. Based on the latest DSP technology, they offer unprecedented performance.

Suitable for control of brushless DC, brushless AC (vector control), DC brushed motors and step motors, the TMC-3D drives accept as position feedback quadrature incremental encoders, absolute encoders (SSI for brushless AC or DC brushed motors).

The TMC-3D can control up to 8 axes at a time. The slave drives can perform multi axis coordinated position control, speed or torque control and work in multi-axis configurations. Thanks to the embedded multi-axis motion controller, the TMC-3D combines multi axis controller with drive in a single compact unit and is capable to execute complex motions without requiring intervention of an external host device. Using the high-level Technosoft Motion Language (**TML**) the following operations can be executed directly at motion controller level:

- Setting multi axis motion modes (Linear Interpolation on 2 or 3 axes, Vector Mode with Tangent Axis)
- Setting various motion modes on all slaves (position profiles, speed profiles etc.)
- Changing the motion modes and/or the motion parameters
- Executing homing sequences on all slaves
- Powerful mechanism of error handling with the possibility of saving the errors that occurs on the slaves in the E2ROM memory
- Controlling program flow through:
  - Conditional jumps and calls of TML functions
  - TML interrupts generated on pre-defined or programmable conditions (protections triggered, transitions on limit switch or capture inputs, etc.)
  - Waits for programmed multi-axis events to occur
- Handling of digital I/O and analogue input signals
- Executing arithmetic and logic operations
- Performing data transfers between motion controller and slaves
- Synchronizing all the axes from the network

Using **EasyMotion Studio** for TML programming you can develop complex multi-axis applications, reducing both the development time and the overall communication requirements. For example, you can command a movement of a slave and with the help of the multi-axis events the slave will inform the master when the task is done.

The TMC-3D can also be controlled from a PC using the family of **TML_LIB** motion libraries.

For all motion programming options, the TMC-3D commissioning for your application is done using **EasyMotion Studio**.

2.2. Key Features

- Built-in multi-axis motion controller and digital drive for control of brushless DC, brushless AC, DC brushed and step motors
- Position, speed or torque control
- Programmable with high-level TML motion language
- Various motion programming modes:
  - Linear Interpolation on 2 or 3 axes
  - Vector Mode with the possibility of adding a third tangent axis
  - Position profiles with trapezoidal or S-curve speed shape
  - External analogue or digital reference
  - 35 Homing modes
- Incremental encoder and digital Hall sensors interfaces: 5V single-ended, open-collector or RS-422 differential
- Absolute SSI encoder interface: RS-422 differential
- Digital I/Os:
  - 6 inputs 24V, opto-isolated, common I/O ground: 2 general-purpose, 2 for limit switches, 2 for Reset and Enable (emergency shutdown)
  - 2 inputs 24V / 5V compatible (shared with second encoder / pulse & direction)
  - 6 digital outputs, opto-isolated, 24V PNP-type, 80/160 mA, short-circuit protected: 4 general-purpose, 2 for Ready and Error
- 2 differential analog inputs +/-10 V, for reference and feedback
- Compact design: 136 x 95 x 26 mm
- RS-232 serial communication up to 115kbaud
- CAN-bus 2.0B up to 1Mbit/s, opto-isolated, using TMLCAN communication protocol compatible with all Technosoft drives with CANbus interface
- Motor temperature sensor interface
- 4K×16 SRAM for data acquisitions and 8K×16 E^2ROM for setup data and TML programs
- Nominal PWM switching frequency[1]: 20 kHz
- Nominal update frequency for torque loop[1]: 10 kHz
- Update frequency for speed/position loop[2]: 1-10 kHz
- Continuous output current: $8A_{RMS}$
- Peak output current: 16.5A
- Logic power supply: 12÷48 V_{DC}
- Motor power supply: 12÷80 V_{DC}
- Minimal load inductance: 50μH @12V, 200 μH @ 48 V, 330 μH @80V
- Operating ambient temperature[3]: 0-40°C

[1] Nominal values cover all cases. Higher values are possible in specific configurations. For details contact Technosoft

[2] 1-2kHz cover all cases. Higher values equal with torque loop update frequency are possible with quadrature encoders

[3] For higher ambient temperatures, contact Technosoft to get derating information

2.3. Supported Motor-Sensor Configurations

The drive part of TMC-3D supports the following configurations:

1. Position, speed or torque control of a **brushless AC rotary motor** with an **incremental quadrature encoder** on its shaft. The brushless motor is vector controlled like a permanent magnet synchronous motor. It works with **sinusoidal** voltages and currents. Scaling factors take into account the transmission ratio between motor and load (rotary or linear). Therefore, the motion commands (for position, speed and acceleration) expressed in SI units (or derivatives) refer to the load[1], while the same commands, expressed in IU units, refer to the motor.

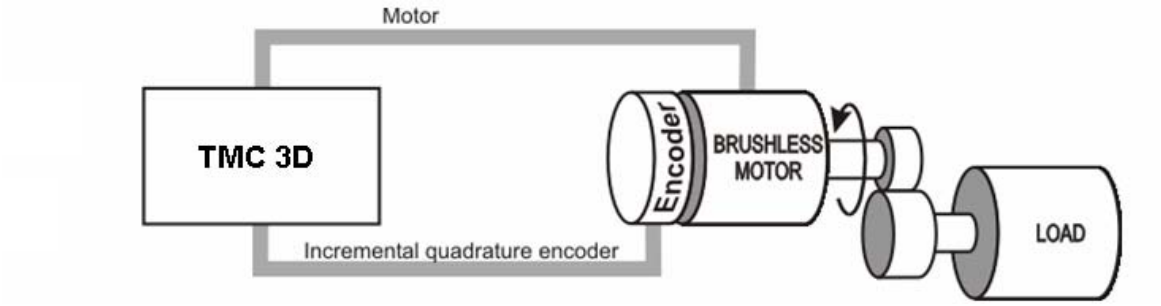


Figure 2.1. *Brushless AC rotary motor. Position/speed/torque control. Quadrature encoder on motor.*

2. Position, speed or torque control of a **brushless AC linear motor** with an **incremental quadrature encoder**. The brushless motor is vector controlled like a permanent magnet synchronous motor. It works with **sinusoidal** voltages and currents. Scaling factors take into account the transmission ratio between motor and load (rotary or linear). Therefore, the motion commands (for position, speed and acceleration) expressed in SI units (or derivatives) refer to the load, while the same commands, expressed in IU units, refer to the motor.

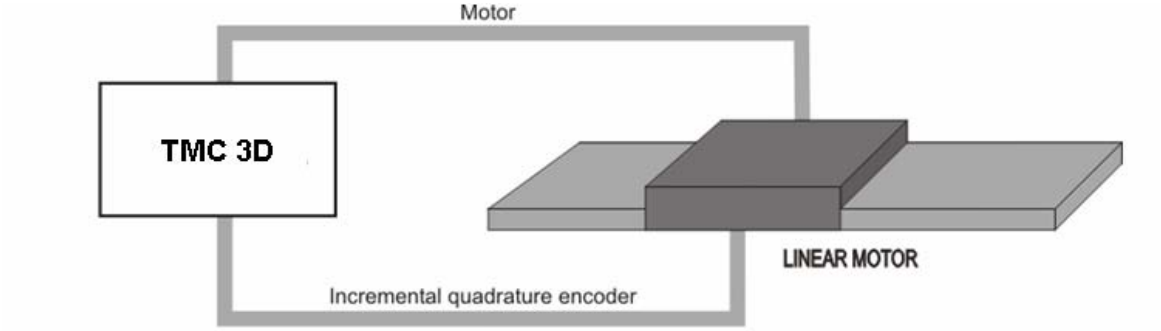


Figure 2.2. *Brushless AC linear motor. Position/speed/torque control. Quadrature encoder on motor.*

[1] Motion commands can be referred to the motor by setting in EasySetUp a rotary to rotary transmission with ratio 1:1

3. Position, speed or torque control of a **brushless DC rotary motor** with **digital Hall sensors** and an **incremental quadrature encoder** on its shaft. The brushless motor is controlled using Hall sensors for commutation. It works with rectangular currents and **trapezoidal** BEMF voltages. Scaling factors take into account the transmission ratio between motor and load (rotary or linear). Therefore, the motion commands (for position, speed and acceleration) expressed in SI units (or derivatives) refer to the load[1], while the same commands, expressed in IU units, refer to the motor.

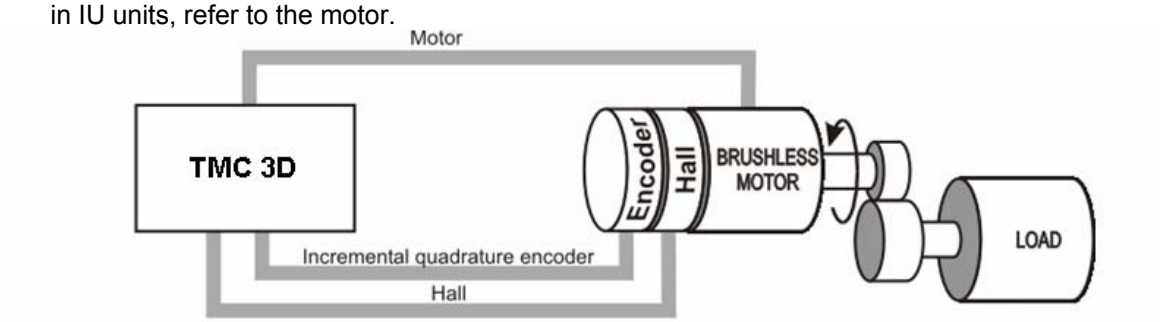


Figure 2.3. *Brushless DC rotary motor. Position/speed/torque control. Hall sensors and quadrature encoder on motor*

4. Position, speed or torque control of a **brushless DC linear motor** with **digital Hall sensors** and an **incremental quadrature encoder**. The brushless motor is controlled using Hall sensors for commutation. It works with rectangular currents and **trapezoidal** BEMF voltages. Scaling factors take into account the transmission ratio between motor and load (rotary or linear). Therefore, the motion commands (for position, speed and acceleration) expressed in SI units (or derivatives) refer to the load, while the same commands, expressed in IU units, refer to the motor.

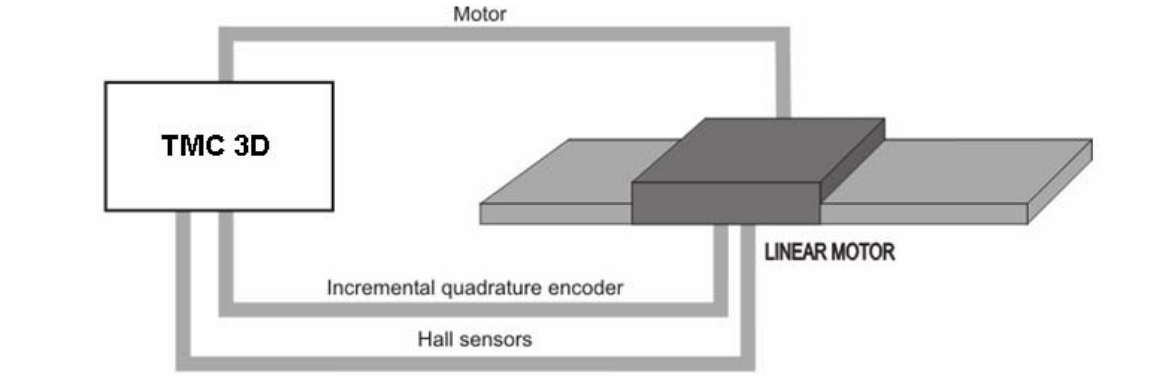


Figure 2.4. *Brushless DC linear motor. Position/speed/torque control. Hall sensors and quadrature encoder on motor*

5. Position, speed or torque control of a **brushless AC rotary motor** with an **absolute SSI encoder** on its shaft. The brushless motor is vector controlled like a permanent magnet

[1] Motion commands can be referred to the motor by setting in EasySetUp a rotary to rotary transmission with ratio 1:1

synchronous motor. It works with **sinusoidal** voltages and currents. Scaling factors take into account the transmission ratio between motor and load (rotary or linear). Therefore, the motion commands (for position, speed and acceleration) expressed in SI units (or derivatives) refer to the load[1], while the same commands, expressed in IU units, refer to the motor.

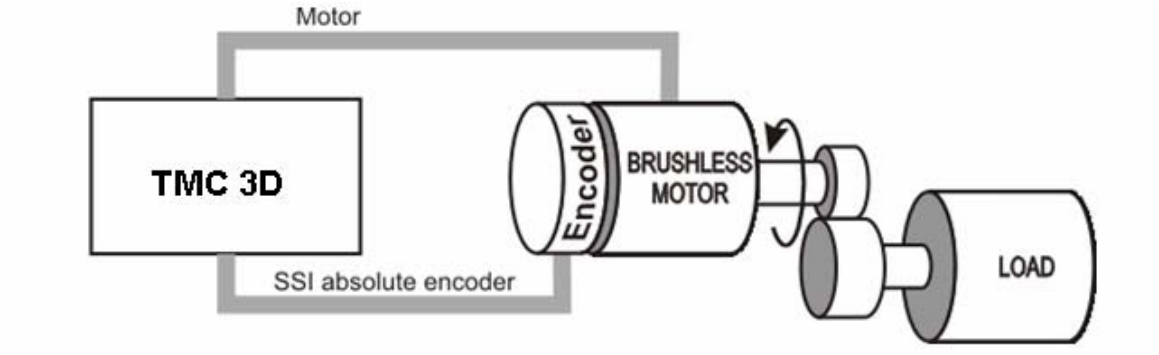


Figure 2.5. *Brushless AC rotary motor. Position/speed/torque control. SSI encoder on motor*

6. Position, speed or torque control of a **DC brushed rotary motor** with an **incremental quadrature encoder** on its shaft. Scaling factors take into account the transmission ratio between motor and load (rotary or linear). Therefore, the motion commands (for position, speed and acceleration) expressed in SI units (or derivatives) refer to the load[1], while the same commands, expressed in IU units, refer to the motor.

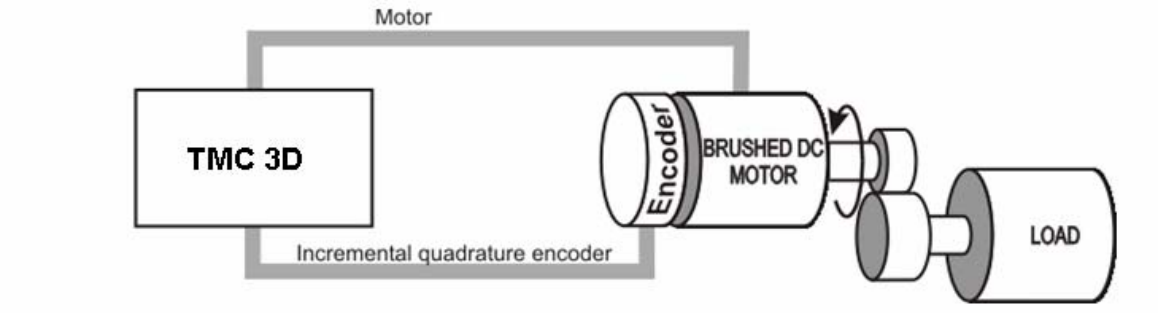


Figure 2.6. *DC brushed rotary motor. Position/speed/torque control. Quadrature encoder on motor*

7. Load position control using an **incremental quadrature encoder** on load, combined with speed control of a **DC brushed rotary motor** having a **tachometer** on its shaft. The motion commands (for position, speed and acceleration) in both SI and IU units refer to the load

[1] Motion commands can be referred to the motor by setting in EasySetUp a rotary to rotary transmission with ratio 1:1

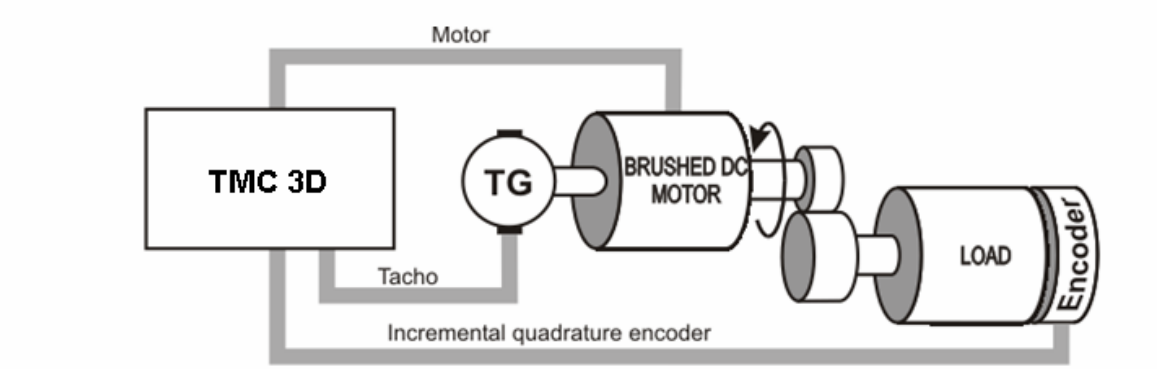


Figure 2.7. *DC brushed rotary motor. Position/speed/torque control. Quadrature encoder on load plus tachometer on motor*

8. Load position control using an **absolute SSI encoder** on load, combined with speed control of a **DC brushed rotary motor** having a **tachometer** on its shaft. The motion commands (for position, speed and acceleration) in both SI and IU units refer to the load

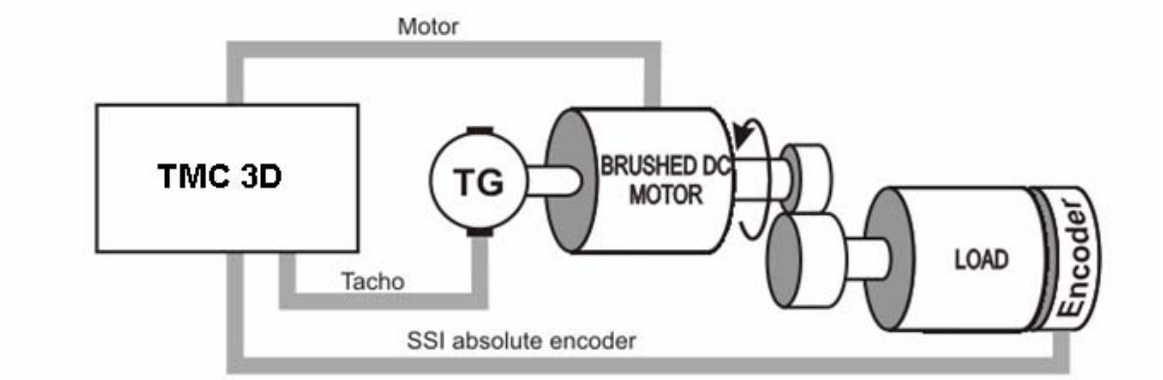


Figure 2.8. *DC brushed rotary motor. Position/speed/torque control. Absolute SSI encoder on load plus tachometer on motor*

9. Open-loop control of a 2 or 3-phase **step motor** in position or speed. Scaling factors take into account the transmission ratio between motor and load (rotary or linear). Therefore, the motion commands (for position, speed and acceleration) expressed in SI units (or derivatives) refer to the load, while the same commands, expressed in IU units, refer to the motor.

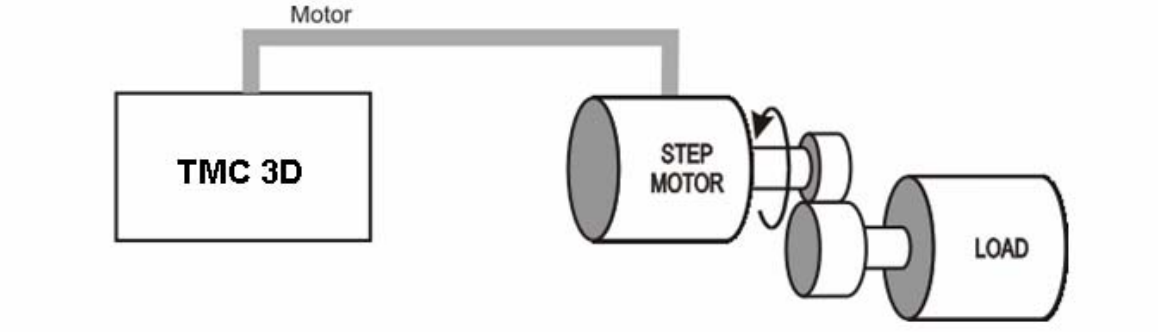


Figure 2.9. *No position or speed feedback. Open-loop control: motor position or speed .*

10. Closed-loop control of **load position using an encoder on load**, combined with open-loop control of **a 2 phase step motor** in speed, with speed reference provided by the position controller. The motion commands in both SI and IU units refer to the load.

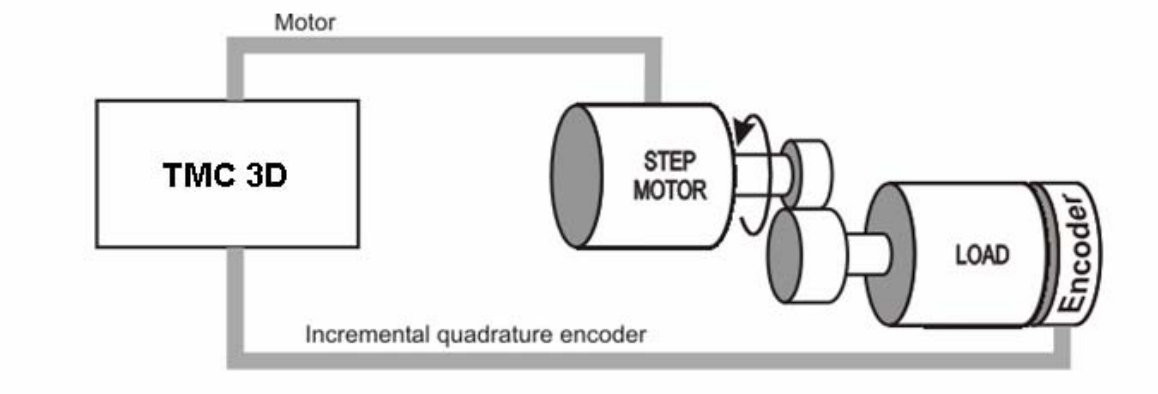


Figure 2.70. *Encoder on load. Closed-loop control: load position, open-loop control: motor speed*

11. Closed-loop control of **a 2-phase step motor** in position, speed or torque. Scaling factors take into account the transmission ratio between motor and load (rotary or linear). Therefore, the motion commands expressed in SI units (or derivatives) refer to the load[1], while the same commands, expressed in IU units refer to the motor.

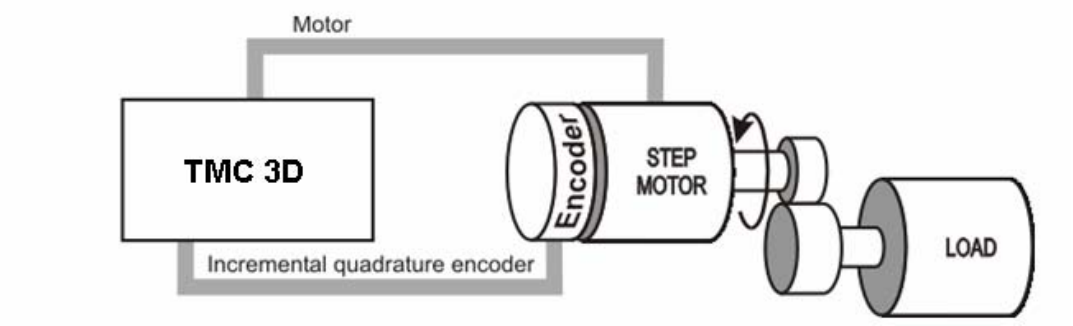


Figure 2.81. *Encoder on motor shaft. Closed-loop control: motor position, speed or torque*

2.4. TMC 3D Dimensions

The next figure presents the TMC 3D drives dimensions.

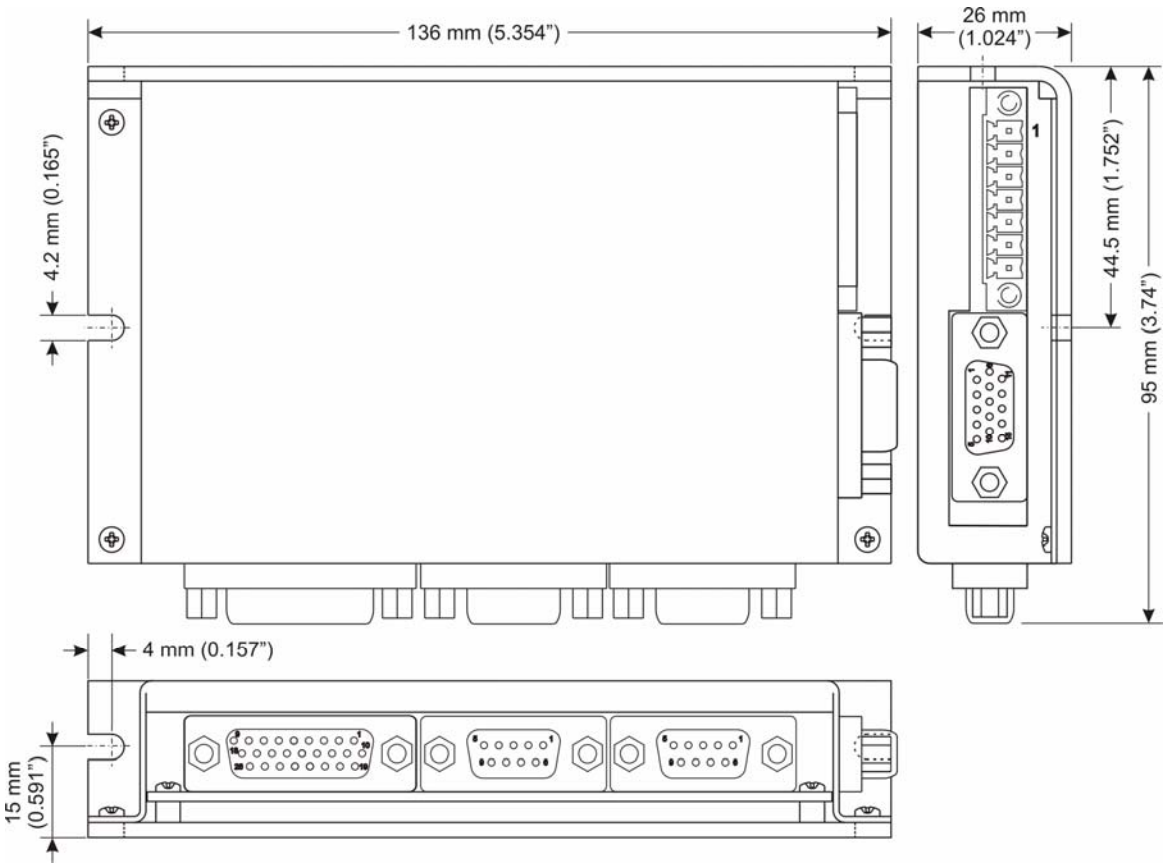


Figure 2.9. *TMC 3D drives dimensions*

2.5. Electrical Specifications

All parameters measured under the following conditions (unless otherwise noted):

T_{amb} = 0…40°C, V_{LOG} = 24 V_{DC}, $V_{24\ VPLC}$ = 24 V_{DC}, CAN_V+ = 24 V_{DC}; V_{MOT} = 80V_{DC}; Load current = 8A_{RMS}; Supplies start-up / shutdown sequence: -*any*-;

Logic Supply Input

	Measured between +V_{LOG} and GND.	Min.	Typ.	Max.	Units
Supply voltage	Nominal values, including ripple	12	24	48	V_{DC}
	Absolute maximum values, continuous	8		+51	V_{DC}

	Absolute maximum values, surge (duration ≤ 10ms)†	-100		+60	V
Supply current	$+V_{LOG}$ = 12V		350	400	mA
	$+V_{LOG}$ = 24 V		150	250	mA
	$+V_{LOG}$ = 48 V		80	150	mA

Motor Supply Input

	Measured between $+V_{MOT}$ and GND.	Min.	Typ.	Max.	Units
Supply voltage	Nominal values, including ripple & braking-induced over-voltage up to ±25%	12		80	V_{DC}
	Absolute maximum values, continuous	0		100	V_{DC}
	Absolute maximum values, surge (duration ≤ 10ms)†	-0.5		105	V
Supply current	Idle		0.5	1.5	mA
	Operating			16.5	A
	Absolute maximum values, surge (duration ≤ 10ms)†			100	A

I/O Supply Input (isolated)

	Measured between +24 V_{PLC} and 0V_{PLC}.	Min.	Typ.	Max.	Units
Supply voltage	Nominal values	8	24	30	V_{DC}
	Absolute maximum values, surge (duration ≤ 10ms)†	-100		32	V
Supply current	All inputs and outputs disconnected		20	30	mA
	All inputs tied to +24 V_{PLC}; all outputs sourcing simultaneously their nominal current into external load(s)		700	1000	mA
Isolation voltage rating	Between 0V_{PLC} and GND			200	V_{RMS}

CAN-Bus Supply Input (isolated)

	Measured between CAN_V+ and CAN_GND.	Min.	Typ.	Max.	Units
Supply voltage	Nominal values	8	24	30	V_{DC}
	Absolute maximum values, surge (duration ≤ 10ms)†	-75		32	V
Supply current	CAN-Bus idle		12	25	mA
	CAN-Bus operating at 1Mbit/s		60	180	mA
Isolation voltage rating	Between CAN_GND and drive GND			200	V_{RMS}

Motor Outputs

	All voltages referenced to GND.	Min.	Typ.	Max.	Units
Motor output current	Continuous operation	-8		+8	A_{RMS}
Motor output current, peak		-16.5		+16.5	A
Short-circuit protection threshold		±26	±25	±29	A
Short-circuit protection delay		12	15		µS
On-state voltage drop	Output current = ±8 A	-1100	±250	+600	mV
Off-state leakage current		-1	±0.1	+1	mA
Motor inductance	F_{PWM} = 20 kHz, $+V_{MOT}$ = 12 V	50			µH
	F_{PWM} = 20 kHz, $+V_{MOT}$ = 48 V	200			µH
	F_{PWM} = 20 kHz, $+V_{MOT}$ = 80 V	400			µH

24 V Digital Inputs (opto-isolated)

	All voltages referenced to $0V_{PLC}$.	Min.	Typ.	Max.	Units
Input voltage	Logic "LOW"	-5	0	1.2	V
	Logic "HIGH"	8	24	30	
	Absolute maximum, surge (duration ≤ 1s) †	-30		+80	
Input current	Logic "HIGH"	2.5	10	15	mA
	Logic "LOW"	0		0.2	
Input frequency		0		5	kHz
Minimum pulse width	Pulse "LOW"-"HIGH"-"LOW"	10			µs
	Pulse "HIGH"-"LOW"-"HIGH"	100			µs

Pulse / Direction / Master Encoder Inputs

		Min.	Typ.	Max.	Units
Single-Ended mode compliance	IN+; Leave IN- disconnected	TTL / CMOS / open-collector 24V referenced to GND			
	IN-; Leave IN+ disconnected				
Differential Mode Compliance	Both IN+, IN- driven; for full RS-422 compliance, see [1]	TIA / EIA – 422			
Input voltage	IN+; Logic "LOW"	-7	0	1.2	V
	IN+; Logic "HIGH"	1.8	5	12	
	IN-; Logic "LOW"	-7	0	4.6	
	IN-; Logic "HIGH"	5.4	24	30	
	Absolute maximum, surge (duration ≤ 1s) †	-12		32	
	Differential input hysteresis	±0.1	±0.2	±0.4	

	Common-mode range (differential input mode)	-12	-7… …12	30	
Input impedance	IN+		1		kΩ
	IN-		0.77		
	Differential impedance[1]	1.5			
Input frequency	Single-ended mode	0		1	MHz
	Differential mode	0		8	MHz
ESD protection	Human body model			±2	kV

24 V Digital Outputs (opto-isolated)

	All voltages referenced to $0V_{PLC}$.	Min.	Typ.	Max.	Units
Output voltage	Logic "HIGH"; $+24\ V_{PLC} = 24\ V_{DC}$; External load = 330Ω	22	23	24.5	V
	Absolute maximum, surge (duration ≤ 1s)†	-0.5		35	
Output current	Logic "HIGH"; $[+24\ V_{PLC} - V_{OUT}] \leq 2$ V; all outputs except $\overline{\text{OUT5}}$ /RD and $\overline{\text{OUT4}}$ /ER			80	mA
	Logic "HIGH"; $[+24\ V_{PLC} - V_{OUT}] \leq 2$ V; outputs $\overline{\text{OUT5}}$ /RD and $\overline{\text{OUT4}}$ /ER			160	mA
	Logic "LOW" (leakage crt.)		0.05	0.2	mA
	Absolute maximum, surge (duration ≤ 1s)†	-350		350	mA

Encoder / Hall Inputs

		Min.	Typ.	Max.	Units
Single-ended mode compliance	Leave negative inputs disconnected	TTL / CMOS / open-collector			
Input threshold voltage	Single-ended mode	1.4	1.5	1.6	V
Differential mode compliance	For full RS422 compliance, see [1]	TIA/EIA-422			
Input hysteresis	Differential mode	±0.1	±0.2	±0.5	V
Input common mode range	Referenced to GND	-7		+12	V
	Absolute maximum, surge (duration ≤ 1s)†	-25		+25	
Input impedance	Single-ended mode		4.7		kΩ
	Differential mode (see [1])		120		Ω
Input Frequency		0		8	MHz
ESD Protection	Human Body Model			±2	kV

SSI Encoder Interface

		Min.	Typ.	Max.	Units
Differential mode compliance (CLOCK, DATA) [1]	For full RS422 compliance, see [1]	TIA/EIA-422			
CLOCK Output voltage	Differential; 50Ω differential load	2.0	2.5	5.0	V
	Common-mode, referenced to GND	2.3	2.5	2.7	
CLOCK frequency	Software selectable	400 to 1500, in 100 increment			kHz
DATA Input hysteresis	Differential mode	±0.1	±0.2	±0.5	V
DATA Input common mode range	Referenced to GND	-7		+12	V
	Absolute maximum, surge (duration ≤ 1s) †	-25		+25	
DATA format	Software selectable	Binary / Gray			
		Single-turn / Multi-turn			
		Counting direction			
DATA resolution	Total resolution			30	bit
	Single-turn			15	bit
	Multi-turn			15	bit

Analog Inputs

		Min.	Typ.	Max.	Units
Differential voltage range			±10		V
Common-mode voltage range	Referenced to GND	-12	0… …10	+50	V
Input impedance	Differential		40		KΩ
Common-mode impedance	Referenced to GND		20		KΩ
Resolution			12		bits
Integral linearity				0.036	% FS[2]
Offset error	Common-mode voltage = 0…10 V		±0.2	±0.5	% FS[2]
Gain error	Common-mode voltage = 0…10 V		±10	±12	% FS[2]
Bandwidth (-3dB)	Depending on software settings		1.5		kHz

RS-232

		Min.	Typ.	Max.	Units
Standards compliance		TIA/EIA-232-C			
Bit rate	Depending on software settings	9600		115200	Baud
ESD Protection	Human Body Model			±15	kV

CAN-Bus

All voltages referenced to CAN_GND		Min.	Typ.	Max.	Units
Standards compliance		CAN-Bus 2.0B error active; ISO 11898-2			
Recommended transmission line impedance	Measured at 1MHz	90	120	150	Ω
Bit rate	Depending on software settings	125K		1M	Baud
Number of network nodes	Bit rate = 125kbps …250kbps			64	-
	Bit rate = 500kbps			50	-
	Bit rate = 1Mbps			32	-
ESD Protection	Human Body Model			±15	kV

Supply Outputs

		Min.	Typ.	Max.	Units
+5 V_{DC} voltage	Current sourced = 350 mA	4.8	5	5.2	V
+5 V_{DC} available current		400	500		mA

Other

		Min.	Typ.	Max.	Units
Operating temperature		0		40	°C
Dimensions	Length x Width x Height	136 x 95 x 26			mm
Weight			0.30		kg
Frame Insulation voltage withstand	GND to SHIELD (connected to frame)			250	V
Storage temperature	Not powered	-40		85	°C
Humidity	Non-condensing	0		90	%RH
Altitude	Referenced to sea-level	0		4000	m
Dust & humidity protection	According to IEC-60925		IP20		

[1] Differential input impedance is ≥1.5KΩ. For full RS-422 compliance, 120Ω termination resistors must be connected across the differential pairs, as close as possible to the drive input pins.

[2] "FS" stands for "Full Scale"

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. Exposure to absolute maximum-rated conditions for extended periods may affect device reliability.

This page is empty

3. Step 1. Hardware Installation

3.1. Mounting

The TMC3D was designed to be cooled by natural convection. It can be mounted horizontally (with label upwards) or vertically inside a cabinet (see Figure 3.1), with motor wires going down. In both cases, leave at least 25mm between the drive and surrounding walls/drives, to allow for free air circulation.

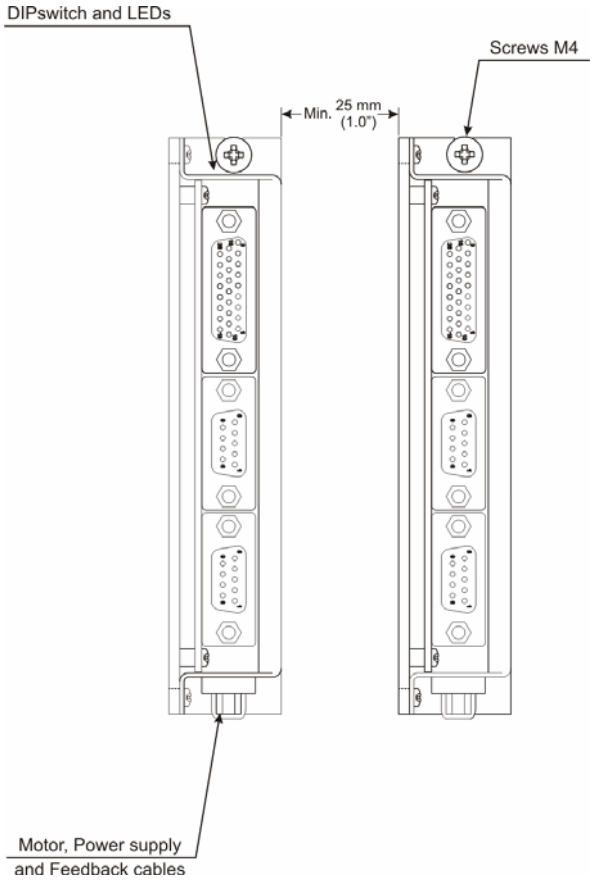


Figure 3.1. *Recommended mounting of TMC 3D in a cabinet*

3.2. Connectors and Connection Diagrams

3.2.1. Connectors Layout

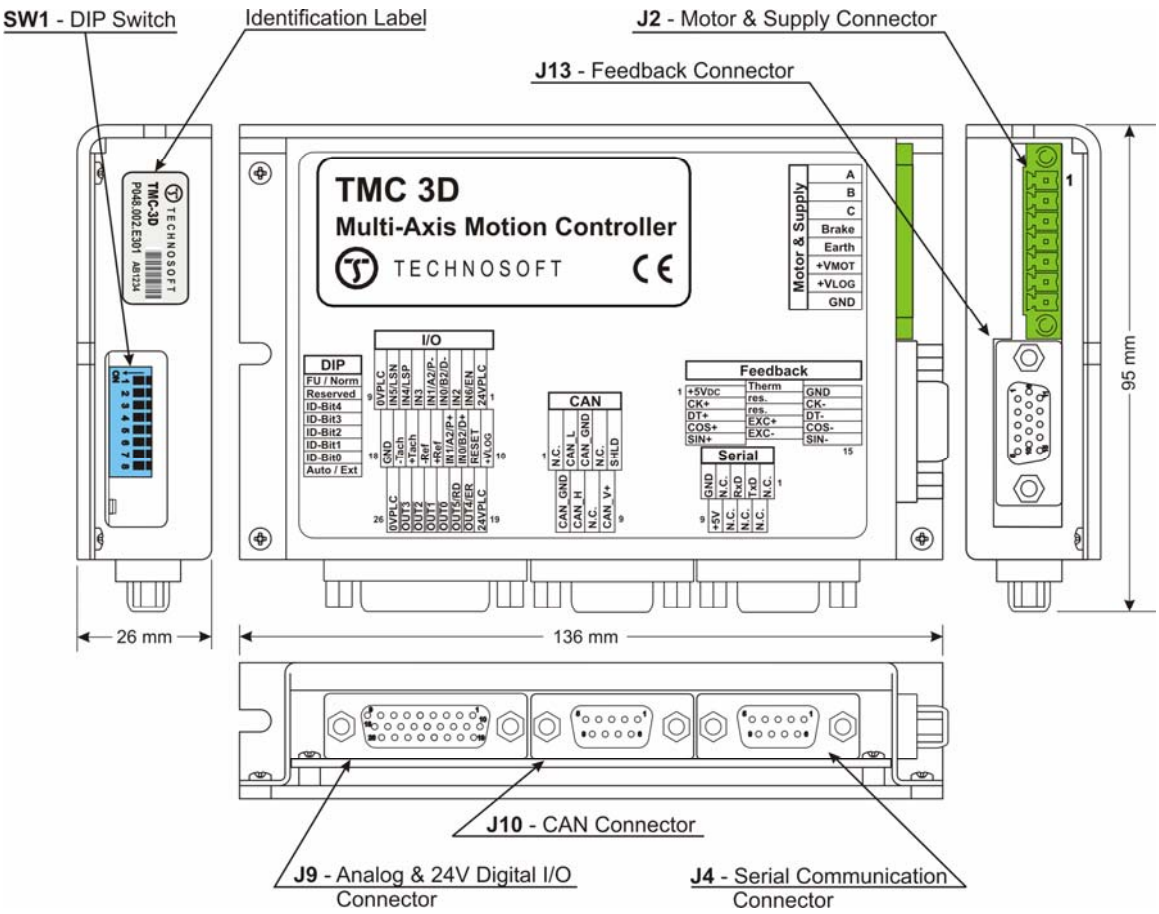


Figure 3.2. *TMC 3D connectors layout*

3.2.2. Identification Labels

Figure 3.3. *TMC 3D Identification Label*

3.2.3. Motor & Supply – J2 Connector

Pin	Name	Type	Function
1	A / A+	O	• Brushless motor or step motor (3-phase): **Phase A** • Step motor (2-phase): Phase **A+** • DC brush motor: **+** (positive terminal)
2	B / A-	O	• Brushless motor or step motor (3-phase): **Phase B** • Step motor (2-phase): **Phase A-** • DC brush motor: **-** (negative terminal)
3	C / B+	O	• Brushless motor or step motor (3-phase): **Phase C** • Step motor (2-phase): **Phase B+** • DC brush motor: not connected
4	BR / B-	O	• **Brake** output for external brake resistor (only when the drive is used with brushless or DC brushed motors) • Step motor (2-phase): **Phase B-** • DC brush motor: not connected
5	Earth	-	Earth connection
6	$+V_{MOT}$	I	Positive terminal of the motor supply: 12 to 80 V_{DC}
7	$+V_{LOG}$	I	Positive terminal of the logic supply: 12 to 48 V_{DC}
8	GND	-	Negative terminal of the $+V_{MOT}$ and $+V_{LOG}$ external power supplies

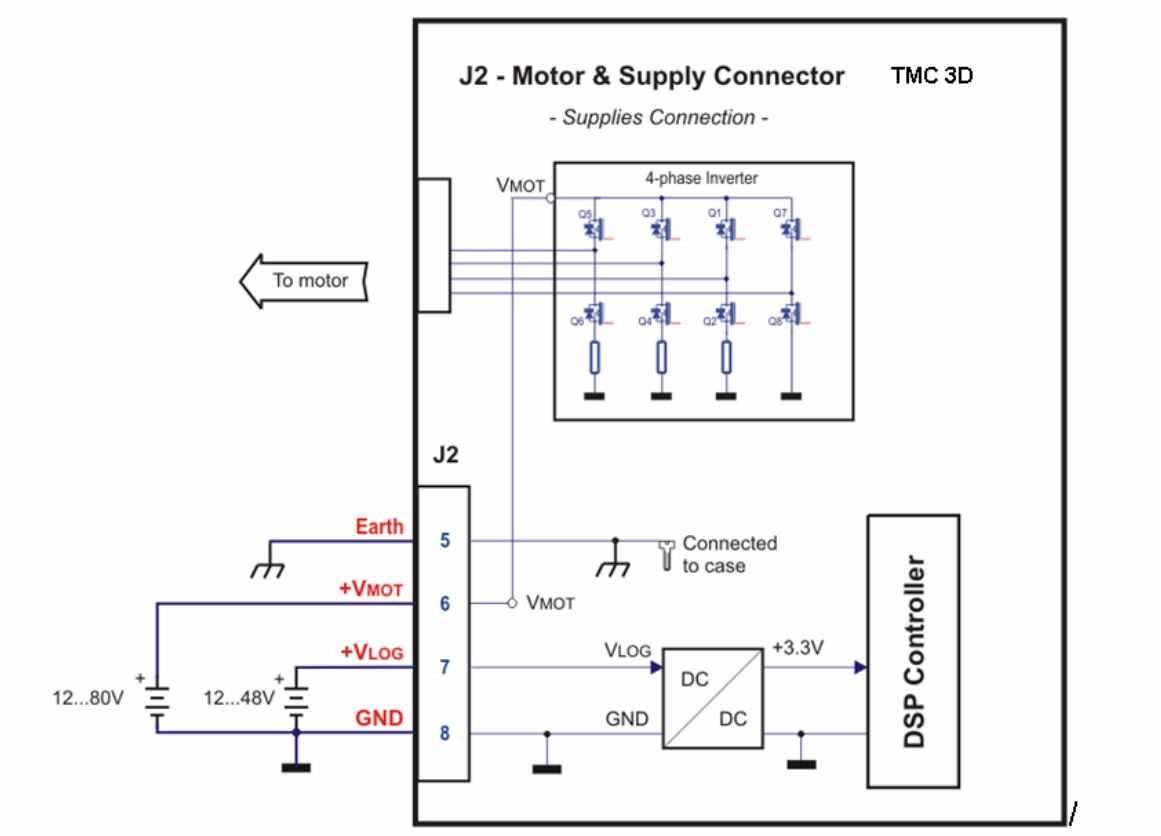


Figure 3.4. *J2 – Supplies connection*

Remark: *The EARTH signal is connected internally to the metal case and to all SHIELD signals. It is completely insulated from all electric signals of TMC 3D. This feature may facilitate avoiding ground loops. It is recommended that Earth be connected to GND at only one point, preferably close to the V_{MOT} supply output.*

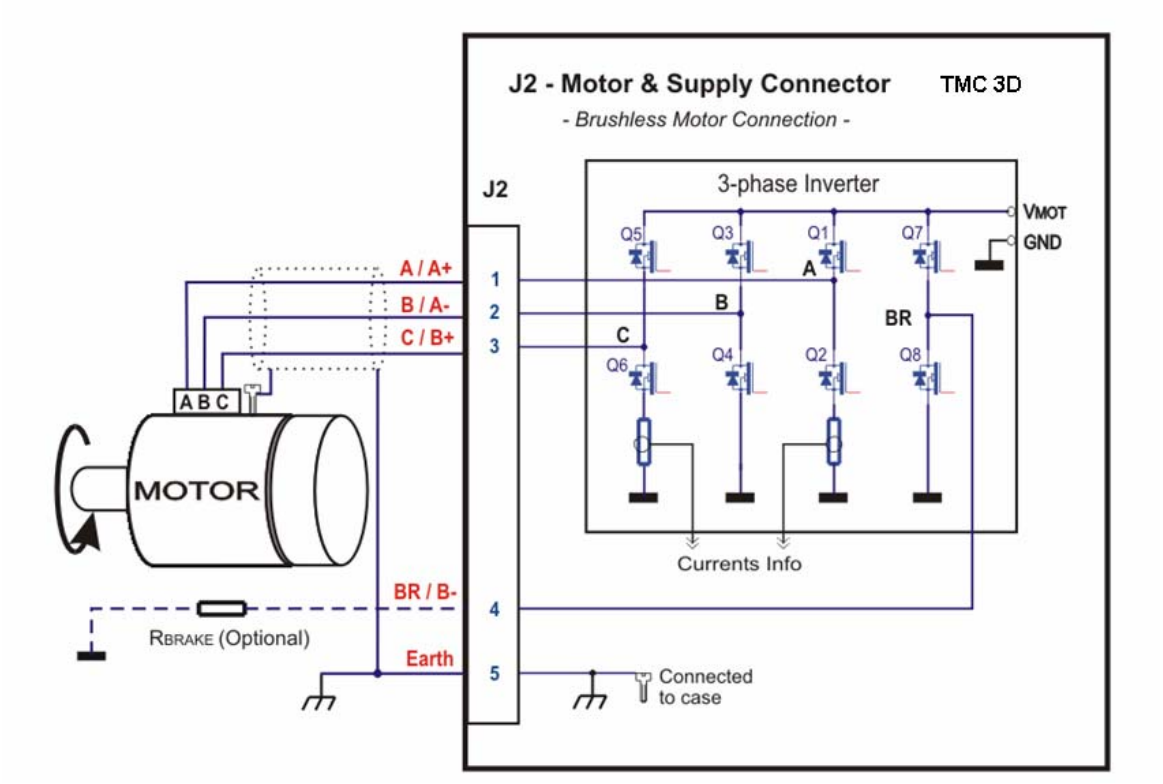


Figure 3.5. *J2 – Brushless motor connection*

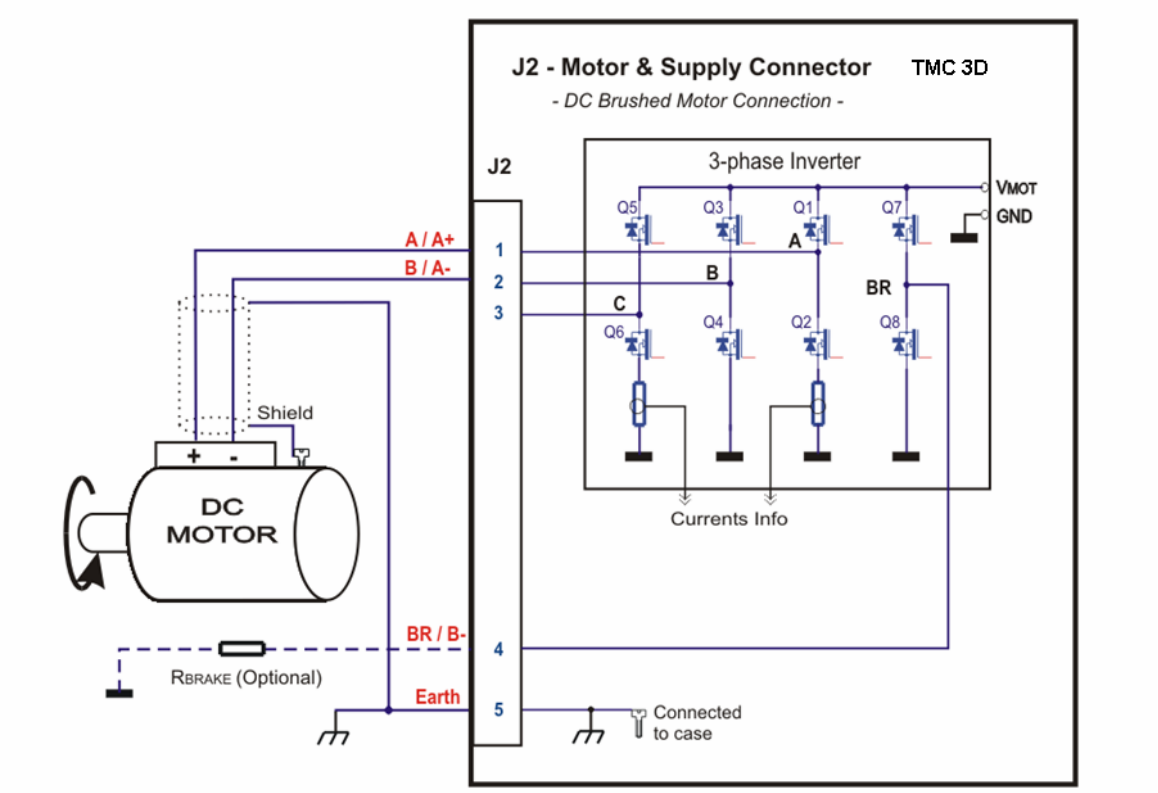


Figure 3.6. *J2 – DC brushed motor connection*

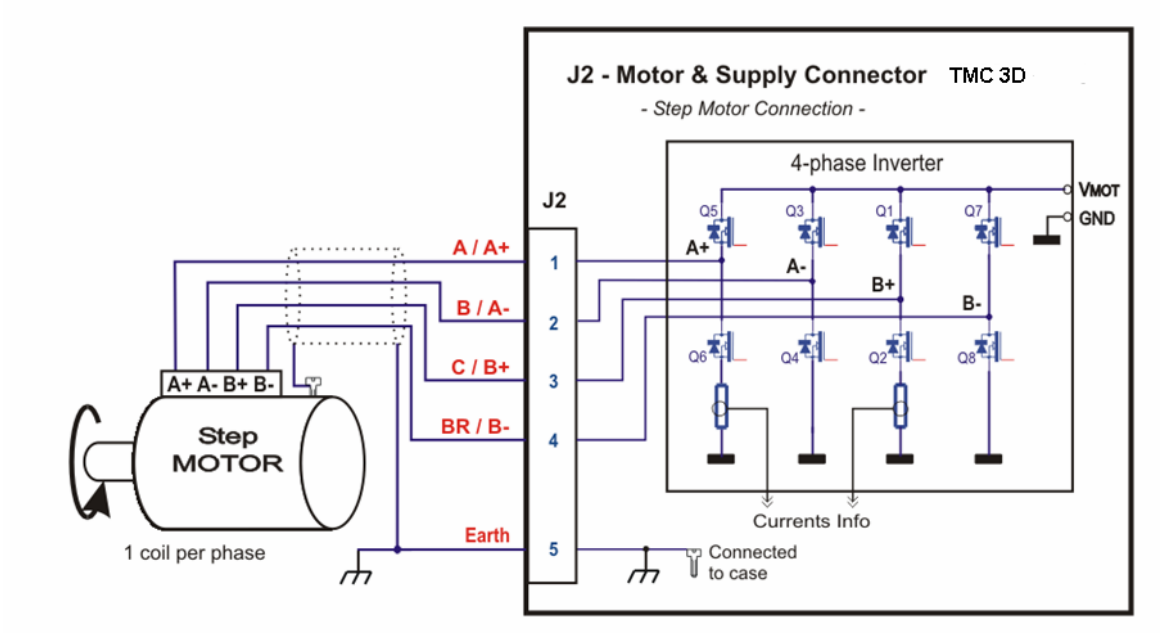


Figure 3.7. *J2 – Step motor connection – 2-phase motor with 1 coil per phase*

Remark: *The EARTH signal is connected internally to the metal case and to all SHIELD signals. It is completely insulated from all electric signals of TMC 3D this feature may facilitate avoiding ground loops. It is recommended that Earth be connected to GND at only one point, preferably close to the V_{MOT} supply output.*

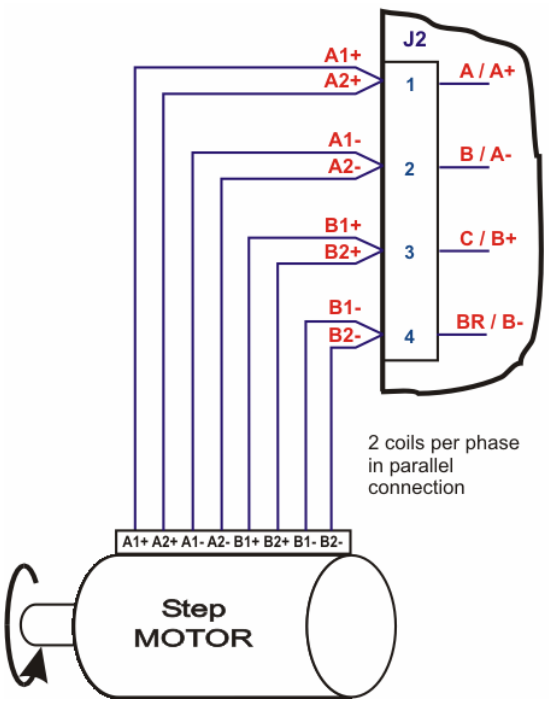


Figure 3.8 *J2 – Connection of a 2-phase motor with 2 coils per phase in parallel*

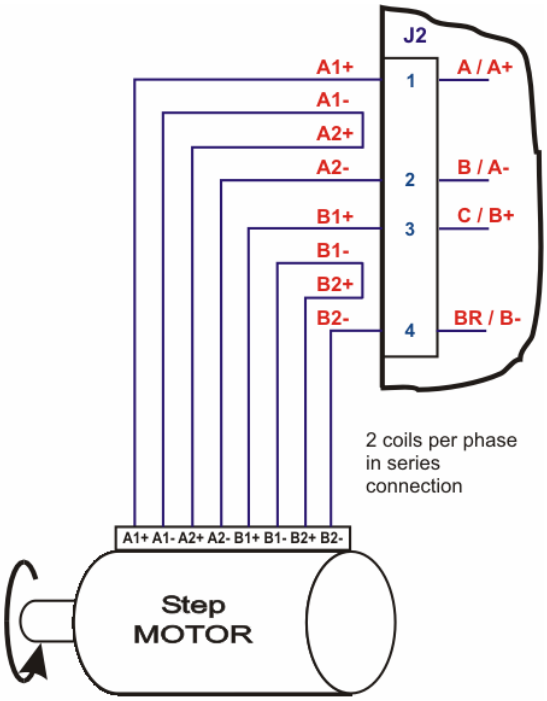


Figure 3.9. *J2 – Connection of a 2-phase motor with 2 coils per phase in series*

3.2.3.1 Recommendations for Motor Wiring

a) Avoid running the motor wires in parallel with other wires for a distance longer than 2 meters. If this situation cannot be avoided, use a shielded cable for the motor wires. Connect the cable shield to the TMC 3D earth/shield pin. Leave the other end disconnected.

b) The parasitic capacitance between the motor wires must not bypass 100nF. If very long cables (hundreds of meters) are used, this condition may not be met. In this case, add series inductors between the TMC 3D outputs and the cable. The inductors must be magnetically shielded (toroidal, for example), and must be rated for the motor surge current. Typically the necessary values are around 100 µH.

c) A good shielding can be obtained if the motor wires are running inside a metallic cable guide.

3.2.3.2 Recommendations for Power Supply On-Off Switch and Wiring

a) If motor supply V_{MOT} is switched on abruptly, the in-rush (start-up) current can reach very high values that can damage the drive. In order to limit the in-rush current, it is preferable to use the inherent soft-start provided by the power supplies when are turned on. Therefore, it is recommended to locate the switch for the motor supply at the INPUT of the power supply (see Figure 3.10) and NOT at the output i.e. between the supply and drive.

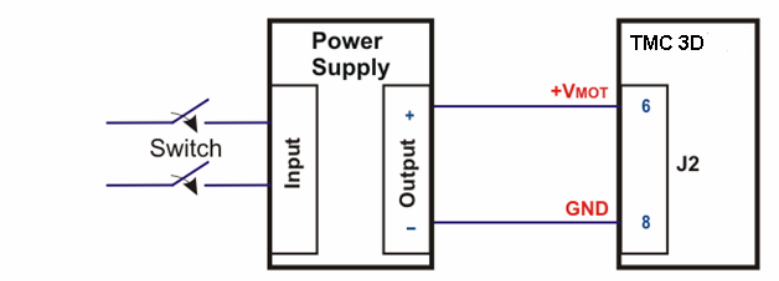


Figure 3.10. *J2 – Motor supply connection – Recommended in-rush current limitation*

b) When the above solution is not possible (as in the case of uninterruptible power supplies, or batteries/accumulators), connect an external capacitor of minimum 470µF between the switch and the drive, to reduce the slew-rate rising slope of the motor supply voltage.

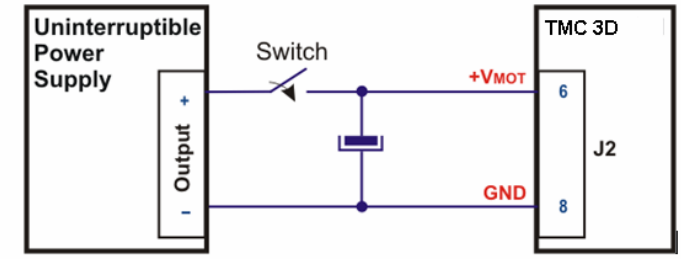


Figure 3.11. *J2 – Motor supply connection – Alternative in-rush current limitation*

WARNING! ***ALWAYS PROVIDE AN EXTERNAL MEAN TO SWITCH OFF THE POWER SUPPLIES! ALWAYS TURN OFF SUPPLIES BEFORE INSTALLING THE DRIVE***

CAUTION! ***ALWAYS LIMIT THE IN-RUSH (START-UP) CURRENT OF THE MOTOR SUPPLY, OTHERWISE IT CAN DAMAGE THE DRIVE***

3.2.3.3 Recommendations for Supply Wiring

1. Use short, thick wires between the TMC 3D and the motor power supply. If the wires are longer than 2 meters, use twisted wires for the supply and ground return. For wires longer than 20 meters, add a capacitor of at least 1,000 µF (rated at an appropriate voltage) right on the terminals of the TMC 3D.
2. When the same motor power supply is used for multiple drives, do a “star” connection centered (electrically) around the supply outputs. Connect each drive to the common motor supply using separate wires for plus and return.
3. Always connect the TMC 3D earth / shield pin to a good quality earth point. The TMC 3D generates electromagnetic disturbances when it’s case is not grounded. Use a short and thick connection from the earth pin of the drive to the earth point. Whenever possible, mount the TMC 3D drive on a metallic surface connected to earth. For mechanical fixing, use good quality plated screws that won't oxidize during the expected lifetime.

3.2.3.4 Recommendations to limit over-voltage during braking

During abrupt motion brakes or reversals the regenerative energy is injected into the motor power supply. This may cause an increase of the motor supply voltage (depending on the power supply characteristics). If the voltage bypasses 92V, the TMC-3D over-voltage protection is triggered and the drive power stage is disabled. In order to avoid this situation you have 2 options:

Option 1. Add a capacitor on the motor supply big enough to absorb the overall energy flowing back to the supply. The capacitor must be rated to a voltage equal or bigger than the maximum expected over-voltage and can be sized with the formula:

$$C \geq \frac{2 \times E_M}{U_{MAX}^2 - U_{NOM}^2} - C_{Drive}$$

where:

U_{MAX} = 92V is the over-voltage protection limit

C_{Drive} = 200 µF is the drive internal capacitance

U_{NOM} = 80V is nominal motor supply voltage

E_M = the overall energy flowing back to the supply in Joules. In case of a rotary motor and load, E_M can be computed with the formula:

$$E_M = \underbrace{\frac{1}{2}(J_M + J_L)\omega_M}_{\text{Kinetic energy}} + \underbrace{(m_M + m_L)g(h_{initial} - h_{final})}_{\text{Potential energy}} - \underbrace{3I_M^2 R_{Ph} t_d}_{\text{Copper losses}} - \underbrace{\frac{t_d \omega_M}{2} T_F}_{\text{Friction losses}}$$

where:

J_M – total rotor inertia [kgm^2]

J_L – total load inertia as seen at motor shaft after transmission [kgm^2]

ω_M – motor angular speed before deceleration [rad/s]

m_M – motor mass [kg] – when motor is moving in a non-horizontal plane

m_L – load mass [kg] – when load is moving in a non-horizontal plane

g – gravitational acceleration i.e. 9.8 [m/s^2]

$h_{initial}$ – initial system altitude [m]

h_{final} – final system altitude [m]

I_M – motor current during deceleration [A_{RMS}/phase]

R_{Ph} – motor phase resistance [Ω]

t_d – time to decelerate [s]

T_F – total friction torque as seen at motor shaft [Nm] – includes load and transmission

In case of a linear motor and load, the motor inertia J_M and the load inertia J_L will be replaced by the motor mass and the load mass measured in [kg], the angular speed ω_M will become linear speed measured in [m/s] and the friction torque T_F will become friction force measured in [N].

Remark: *If the above computation of E_M can't be done due to missing data, a good starting value for the capacitor can be 10,000 µF / 100V.*

Option 2. Connect a brake resistor R_{BR} between pin 4 and pin 8 of the Motor & Supply connector J2 and activate the drive braking circuit from EasySetUp when motor supply voltage exceeds: U_{BRAKE} = 87V. This option is not available when the TMC-3D is used with a step motor.

Remark: *This option can be combined with an external capacitor whose value is not enough to absorb the entire regenerative energy E_M but can help reducing the brake resistor size.*

Brake resistor selection

The brake resistor value must be chosen to respect the following conditions:

1. to limit the maximum current below the TMC-3D peak current I_{PEAK} = 16.5A

$$R_{BR} > \frac{U_{MAX}}{I_{PEAK}}$$

2. to sustain the required *braking power*:

$$P_{BR} = \frac{E_M - \frac{1}{2}C(U_{MAX}^2 - U_{brake}^2)}{t_d}$$

where $C = C_{EXT} + C_{DRIVE}$ is the overall capacitance on the motor supply (external + drive), i.e:

$$R_{BR} < \frac{U_{BRAKE}^2}{2 \times P_{BR}}$$

3. to limit the average current below the TMC-3D nominal current I_{NOM}=8A

$$R_{BR} > \frac{P_{BR} \times t_d}{t_{CYCLE} \times I_{NOM}^2}$$

where t_{CYCLE} is the time interval between 2 brakes in case of repetitive moves.

4. to be rated for an average power $P_{AV} = \frac{P_{BR} \times t_d}{t_{CYCLE}}$ and a peak power $P_{PEAK} = \frac{U_{MAX}^2}{R_{BR}}$

Remarks:

1. *If* $\frac{U_{MAX}}{I_{PEAK}} > \frac{U_{BRAKE}^2}{2 \times P_{BR}}$ *the braking power P_{BR} must be reduced by increasing either t_d – the time to decelerate or C_{EXT} – the external capacitance on the motor supply*
2. *If* $\frac{P_{BR} \times t_d}{t_{CYCLE} \times I_{NOM}^2} > \frac{U_{BRAKE}^2}{2 \times P_{BR}}$ *either the braking power must be reduced (see Remark 1) or t_{CYCLE} – the time interval between braking cycles must be increased*

WARNING! ***THE BRAKE RESISTOR MAY HAVE HOT SURFACES DURING OPERATION.***

3.2.4. Feedback – J13 Connector (TMC 3D)

Pin	Name on the Drive cover	Type	Function / Comments
1	A1+	I	**Positive A** for differential encoder or **A** for single-ended encoder
2	B1+	I	**Positive B** for differential encoder or **B** for single-ended encoder
3	+5 V_{DC}	O	+5 V_{DC} Supply (generated internally)
4	H3/CK+	I/O	**Positive Hall 3** input for differential Hall or **Hall 3** for single-ended Hall **Positive Clock** output signal for differential SSI encoder
5	H1/DT+	I	**Positive Hall 1** for differential Hall or **Hall 1** for single-ended Hall **Positive Data** signal for differential SSI encoder
6	Therm	I	Analog input from motor thermal sensor
7	Z1+	I	**Positive Z** for differential encoder or **Z** for single-ended encoder [1]*)
8	Z1-	I	**Negative Z** for differential encoder
9	H2+	I	**Positive Hall 2** for differential Hall or **Hall 2** for single-ended Hall [2]*)
10	H2-	I	**Negative Hall 2** for differential Hall
11	A1-	I	**Negative A** for differential encoder
12	B1-	I	**Negative B** for differential encoder
13	GND	-	Ground of the encoder supply
14	H3/CK-	I/O	**Negative Hall 3** input for differential Hall; **Negative Clock** output signal for differential SSI encoder
15	H1/DT-	I	**Negative Hall 1** for differential Hall **Negative Data** signal for differential SSI encoder
case	SHIELD	-	Shield; Connected to frame

CAUTION! ***CHECK CURRENT CONSUMPTION FROM +5VDC SUPPLY! BYPASSING THE MAXIMUM ALLOWED CURRENT MIGHT LEAD TO DRIVE MALFUNCTION***

CAUTION! ***THE FEEDBACK CONNECTOR SIGNALS ARE ELECTROSTATICALLY SENSITIVE AND SHALL BE HANDLED ONLY IN AN ESD PROTECTED ENVIRONMENT***

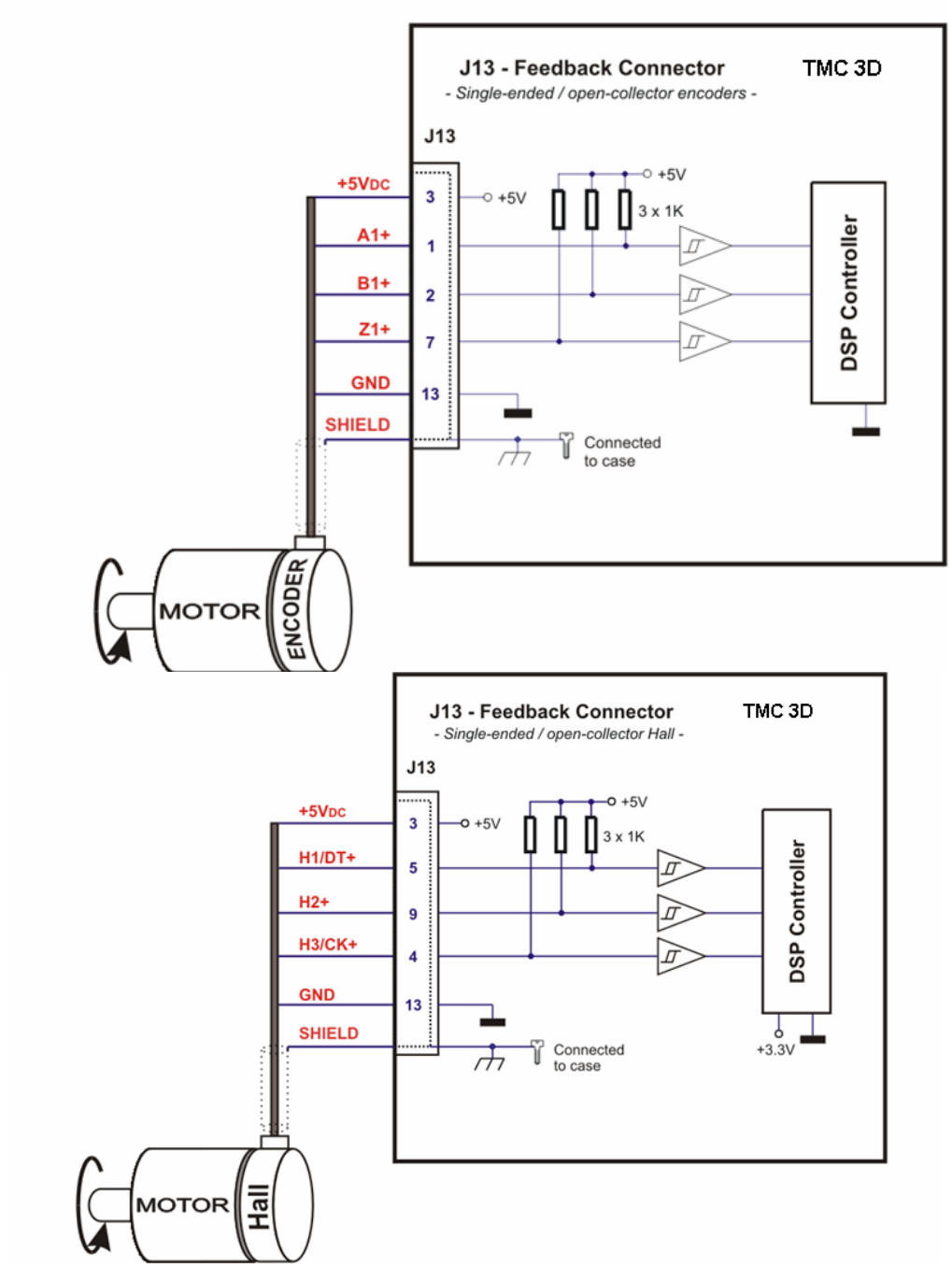


Figure 3.12. *J13 – Single-ended / open-collector encoder and Hall connection*

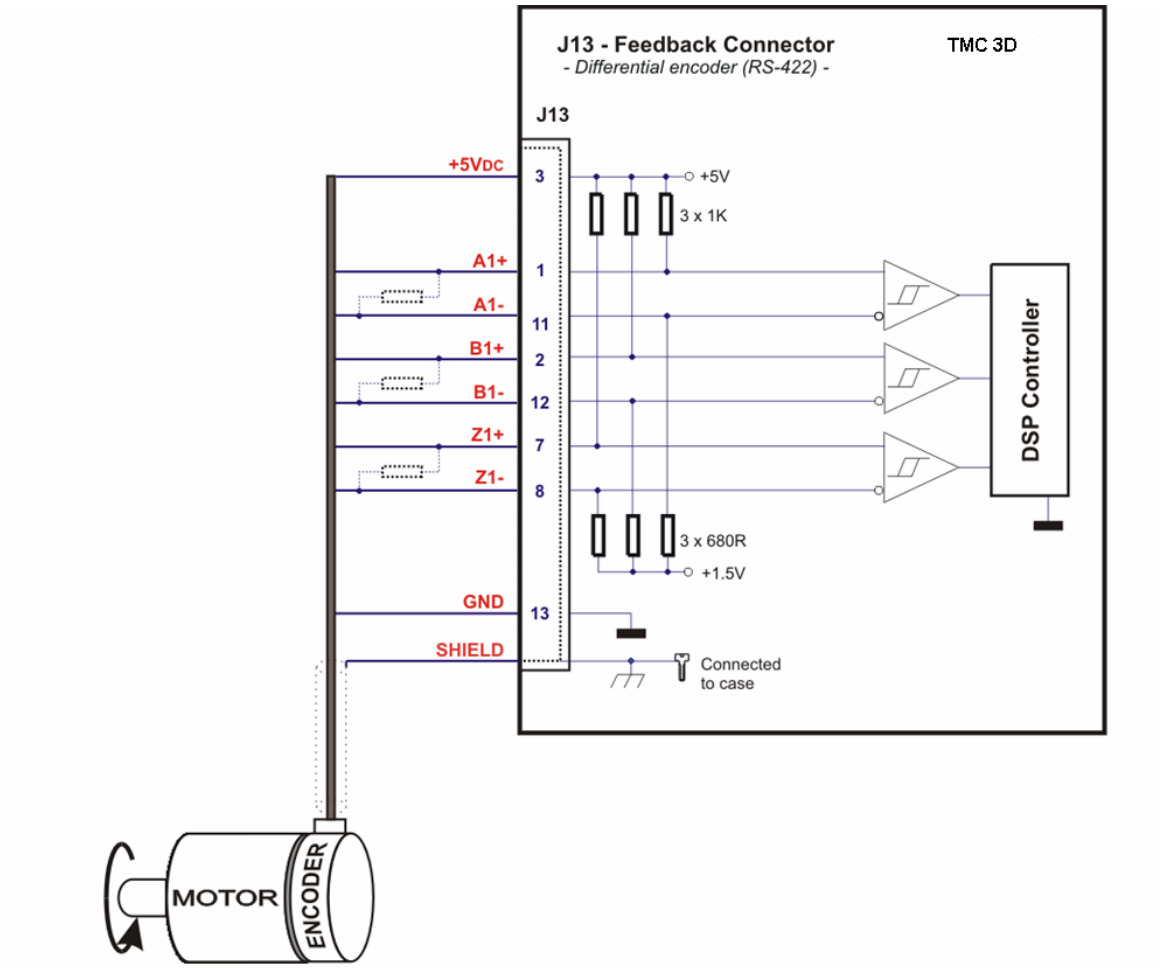


Figure 3.13. *J13 – Differential (RS-422) encoder connection*

Remark: *For long (>10 meters) encoder lines add 120Ω termination resistors close to TMC-3D.*

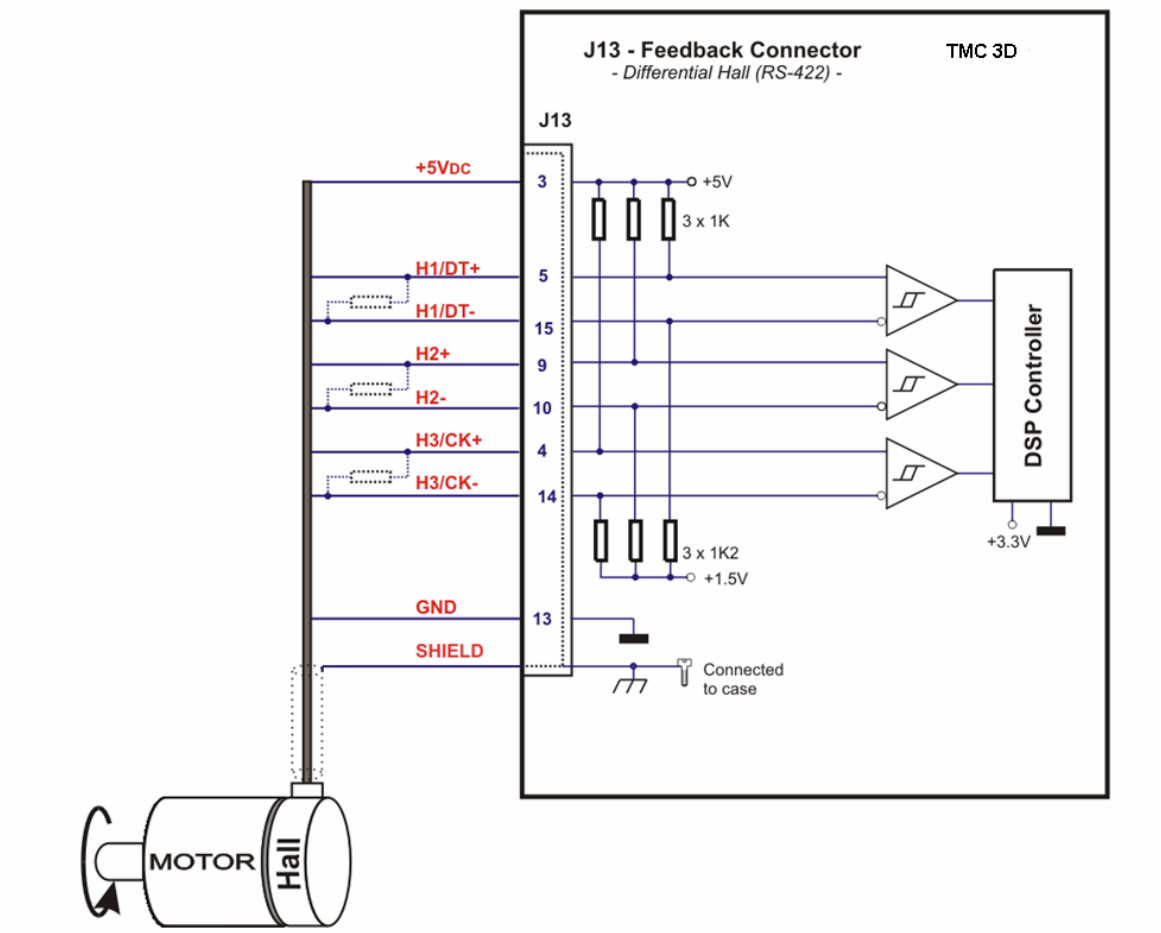


Figure 3.14. *J13 – Differential (RS-422) Hall connection*

Remark: *For long (>10 meters) Hall lines add 120Ω termination resistors close to TMC-3D.*

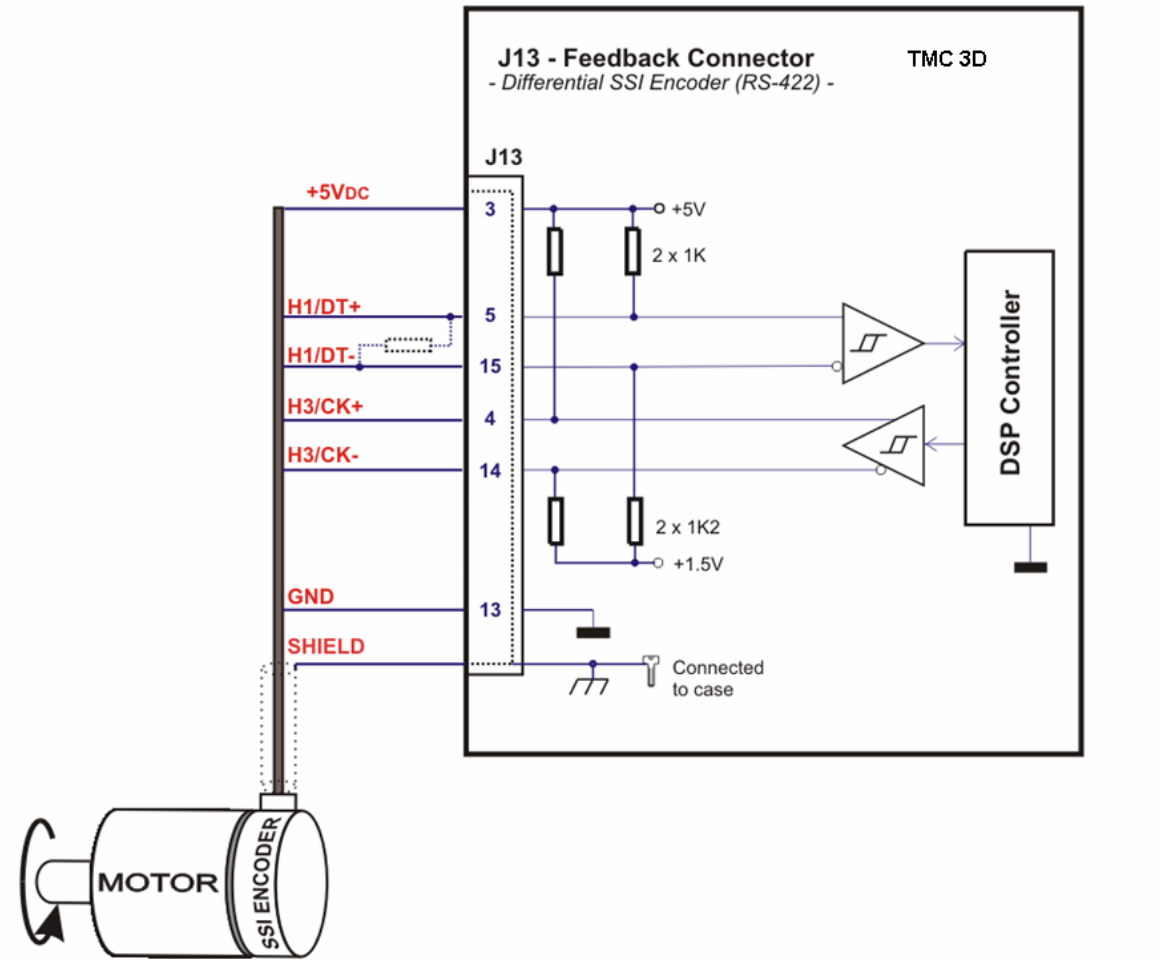


Figure 3.15. *J13 – Differential (RS-422) SSI encoder connection*

Remarks:

1.For long (>10 meters) SSI encoder lines add 120Ω termination resistors close to TMC-3D.

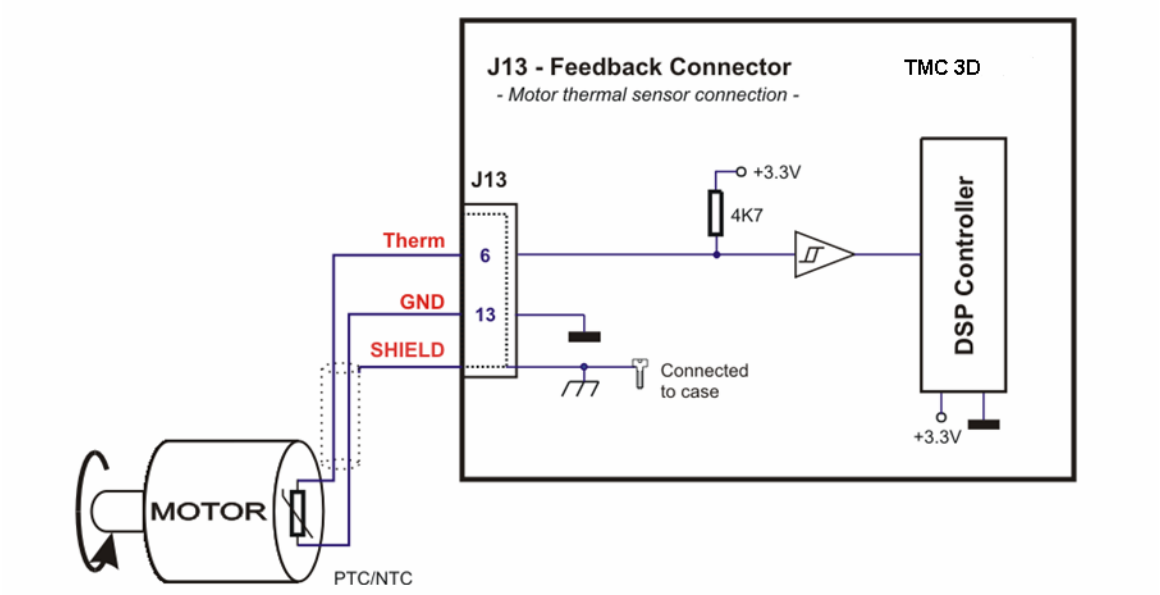


Figure 3.16. *J13 – Motor thermal sensor connection*

3.2.4.1 Recommendations for Feedback Devices Wiring

a) Always connect both positive and negative signals when the encoder or the Hall sensors are differential and provides them. Use one twisted pair for each differential group of signals as follows: A+ with A-, B+ with B-, Z+ with Z-, H1/DT+ with H1/DT-, H2+ with H2-, H3/CK+ with H3/CK-. Use another twisted pair for the 5V supply and GND.

b) Keep the ground connection between an encoder and the TMC-3D even if the encoder supply is not provided by the drive. When using shielded cable, connect the cable shield to the earth at the encoder side. Leave the shield unconnected at the IDS side. **Never use the shield as a conductor caring a signal, for example as a ground line!**. This situation can lead to a worse behavior than a non-shielded cable

c) Always use shielded cables to avoid capacitive-coupled noise when using single-ended encoders or Hall sensors with cable lengths over 1 meter. Connect the cable shield to the earth potential, at only one end. This point could be either the TMC-3D (using the earth/shield pin(s)) or the encoder / motor. Do not connect the shield at both ends.

d) If the TMC-3D 5V supply output is used by another device (like for example an encoder) and the connection cable is longer than 5 meters, add a decoupling capacitor near the supplied device, between the +5V and GND lines. The capacitor value can be 1...10 µF, rated at 6.3V.

3.2.5. Analog & Digital I/O – J9 Connector

Pin	Name on the Drive cover	TML name	Type	Function / Alternate function / Comments
1, 19	24VPLC	-	I	• 24 V power supply (+) terminal for all opto-isolated I/O
2	IN6/EN	IN(6)	I	• 24V Enable input, read as **In(6)**. On inactive level disables the drive operation similarly to AXISOFF command (power stage is turned off). Read high (1 logic) when 24VPLC are applied on IN6/EN pin • Opto-isolated • Programmable polarity / active level
3	IN2/HOME	IN(2)	I	• 24V General-purpose input **In(2)**. Read high (1 logic) when 24VPLC are applied on IN2/HOME pin • 24V Home input in homing sequences. Can be set to capture on transitions both motor and master position • Opto-isolated
4	IN0/B2/D-	IN(0)	I	• RS-422 differential **B-** / 24V single-ended **B** input when external reference is 2nd (master) encoder • RS-422 differential **Dir-** / 24V single-ended **Dir** input when external reference is Pulse & Direction • 24V General-purpose input **In(0)**. Read low (0 logic) when 24VPLC are applied on IN0/B2/D- pin • Compatible RS-422 and 24V single-ended
5	IN1/A2/P-	IN(1)	I	• RS-422 differential **A-** / 24V single-ended **A** input when external reference is 2nd (master) encoder • RS-422 differential **Puls-** / 24V single-ended **Puls** input when external reference is Pulse & Direction • 24V General-purpose input **In(1)**. Read low (0 logic) when 24VPLC are applied on IN1/A2/P- pin • Compatible RS-422 and 24V single-ended
6	IN3	IN(3)	I	• 24V General-purpose input **In(3).** Read high (1 logic) when 24VPLC are applied on IN3 pin • Opto-isolated
7	IN4/LSP	IN(4)	I	• 24V Positive limit switch input. On active level stops motion in positive direction • 24V General-purpose input **In(4)** if limit switches are disabled. Read high (1 logic) when 24VPLC are applied on IN4/LSP pin • Opto-isolated • Programmable polarity / active level

8	IN5/LSN	IN(5)	I	• 24V Negative limit switch input. On active level stops motion in negative direction • 24V General-purpose input **In(5)** if limit switches are disabled. Read high (1 logic) when 24VPLC are applied on IN5/LSN pin • Opto-isolated • Programmable polarity / active level
9, 26	0VPLC	-	-	• Ground (-) terminal for all opto-isolated I/O
10	$+V_{LOG}$	-	O	• $+ V_{LOG}$. Logic supply voltage (as applied on J2, pin 7)
11	RESET	-	I	• RESET pin – connect to 24VPLC to reset the drive
12	IN0/B2/D+	IN(0)	I	• 24V general-purpose input **In(0).** Read high (1 logic) when 24VPLC are applied on IN0/B2/D+ pin • RS-422 differential **B+** / 5V single-ended **B** input when external reference is 2nd (master) encoder • RS-422 differential **Dir+** / 5V single-ended **Dir** input when external reference is Pulse & Direction • Compatible RS-422, 5V and 24V single-ended
13	IN1/A2/P+	IN(1)	I	• 24V general-purpose input **In(1)**. Read high (1 logic) when 24VPLC are applied on IN1/A2/P+ pin • RS-422 differential **A+** / 5V single-ended **A** input when external reference is 2nd (master) encoder • RS-422 differential **Puls+** / 5V single-ended **Puls** input when external reference is Pulse & Direction, or • Compatible RS-422, 5V and 24V single-ended
14	+Ref	AD5	I	• Analogue position, speed or torque reference input • +/-10 V differential • 12-bit resolution
15	-Ref		I	
16	+Tach	AD2	I	• Analogue speed feedback (tachometer input) • +/-10 V differential • 12-bit resolution
17	- Tach		I	
18	GND	-	O	• Ground terminal for all non-isolated I/O
20	$\overline{\text{OUT4}}$ /ER	OUT(4)	O	• 24 V Error output, seen as **Out(4)**. When Out(4) is commanded low (0 logic), $\overline{\text{OUT4}}$ /ER pin is set to +24VPLC and lights the red led • Opto-isolated • Short-circuit protected
21	$\overline{\text{OUT5}}$ /RD	OUT(5)	O	• 24 V Ready output, seen as **Out(5)**. When Out(5) is commanded low (0 logic), $\overline{\text{OUT5}}$ /RD pin is set to +24VPLC and lights the green LED • Opto-isolated • Short-circuit protected

22	$\overline{\text{OUT0}}$	OUT(0)	O	• 24 V General-purpose output **Out(0)**. When Out(0) is commanded low (0 logic), $\overline{\text{OUT0}}$ pin is set to +24VPLC • Opto-isolated • Short-circuit protected
23	$\overline{\text{OUT1}}$	OUT(1)	O	• 24 V General-purpose output **Out(1)**. When Out(1) is commanded low (0 logic), $\overline{\text{OUT1}}$ pin is set to +24VPLC • Opto-isolated • Short-circuit protected
24	$\overline{\text{OUT2}}$	OUT(2)	O	• 24 V General-purpose output **Out(2)**. When Out(2) is commanded low (0 logic), $\overline{\text{OUT2}}$ pin is set to +24VPLC • Opto-isolated • Short-circuit protected
25	$\overline{\text{OUT3}}$	OUT(3)	O	• 24 V General-purpose output **Out(3)**. When Out(3) is commanded low (0 logic), $\overline{\text{OUT3}}$ pin is set to +24VPLC • Opto-isolated • Short-circuit protected
case	SHIELD	-	-	Shield; Connected to frame

CAUTION! ***THE I/O CONNECTOR SIGNALS ARE ELECTRO-STATICALLY SENSITIVE AND SHALL BE HANDLED ONLY IN AN ESD PROTECTED ENVIRONMENT.***

Remarks:

3. *The 24V opto-isolated I/O signals are referenced to the isolated ground 0VPLC, which shall be common to all the devices sharing these signals.*
4. *The 24V opto-isolated inputs have a typical threshold of 8 Volts, therefore will not accept TTL levels.*
5. *The isolated 24VPLC supply is required only for operation of the outputs. Hence, if your application uses only opto-isolated inputs, the 24VPLC supply connection is not necessary.*
6. *The inputs In(0) and In(1) accept both TTL (5V) and 24V signals and are not opto-isolated. These inputs are referenced to the drive logic ground GND*

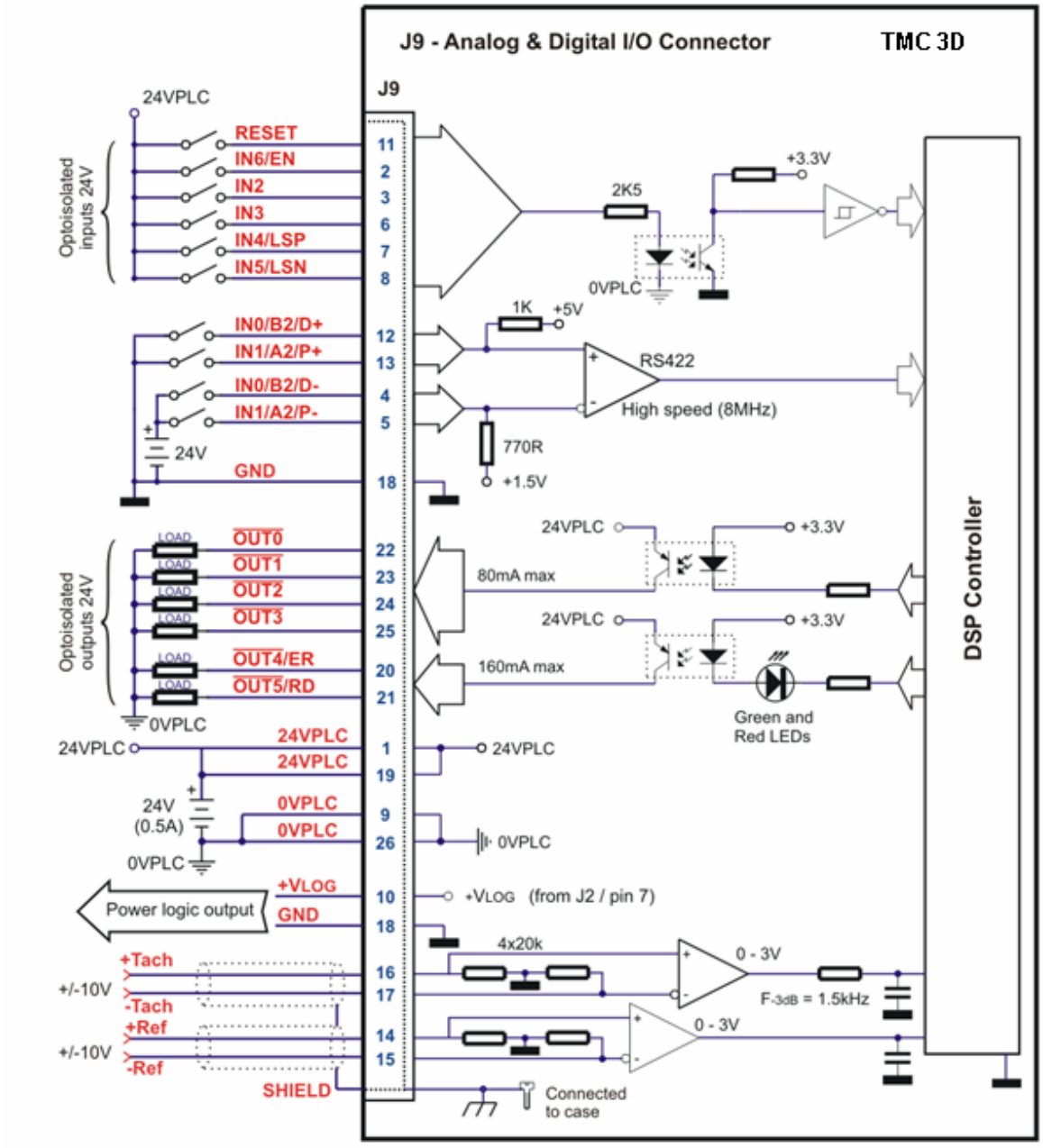


Figure 3.17 *J9 – Analogue & Digital I/O connections*

3.2.5.1 Recommendations for Analogue Signals Wiring

a) If the analogue signal source is single-ended, use a 2-wire shielded cable as follows: 1st wire connects the live signal to the drive positive input (+); 2nd wire connects the signal ground to the drive negative input(-).

b) If the analogue signal source is differential and the signal source ground is isolated from the drive GND, use a 3-wire shielded cable as follows: 1st wire connects the signal plus to the drive positive input (+); 2nd wire connects the signal minus to the drive negative input (-) and 3rd wire connects the source ground to the drive GND

c) If the analogue signal source is differential and the signal source ground is common with the drive GND, use a 2-wire shielded cable as follows: 1st wire connects the signal plus to the drive positive input (+); 2nd wire connects the signal minus to the drive negative input (-)

d) For all of the above cases, connect the cable shield to the drive I/O connector frame and leave the other shield end unconnected to the signal source. To further increase the noise protection, use a double shielded cable with inner shield connected to drive GND and outer shield connected to the drive I/O connector frame. Leave both shields unconnected on the signal source side

e) If the signal source output voltage is larger than +/-10V, use a 3-resistor differential divider, located near the TMC 3D I/O connector. Choose the divider resistances as low as possible, close to the signal source output current limit, to minimize the noise

3.2.6. Serial Communication – J4 Connector

Pin	Name	Type	Function
2	TxD	O	RS-232 Data Transmission
3	RxD	I	RS-232 Data Reception
5	GND	-	Ground
1,4,6,7,8	n.c.	-	Not Connected
9	+5V	O	Optional supply for handheld terminal (internally generated)

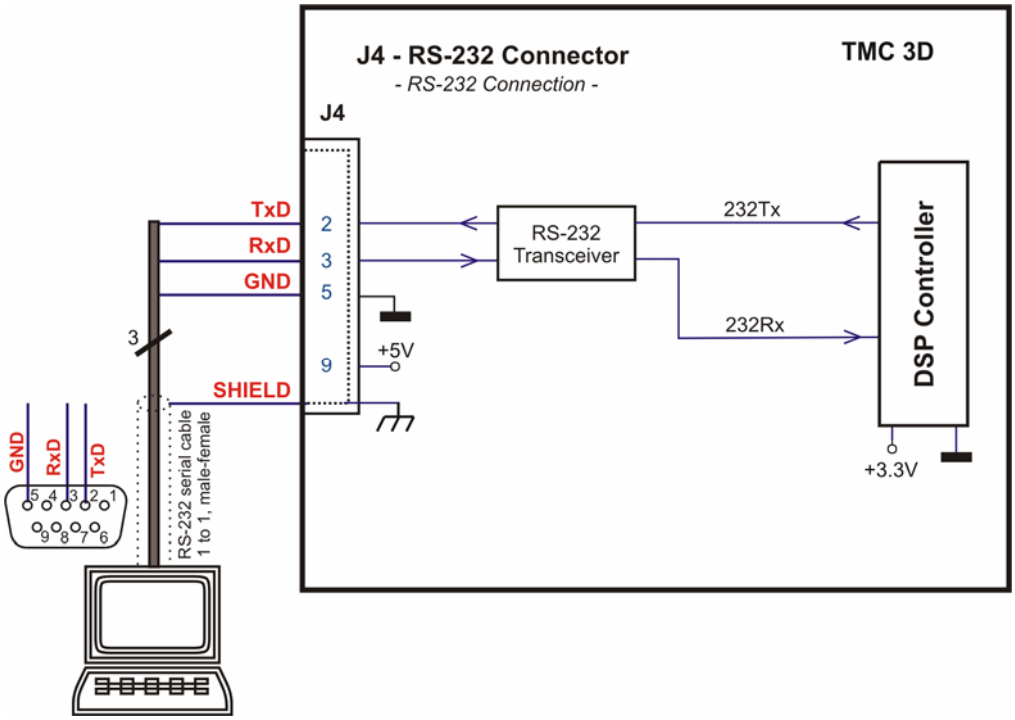


Figure 3.18. *J4 – Serial RS-232 connection*

Remarks:

1. *Use a 9-wire standard 1-to-1 (non-inverting) shielded cable, preferable with metallic or metallized shells (casings)*
2. *On TMC-3D drive the electrical ground (GND) and the earth/shield are isolated*

3.2.6.1 Recommendations for RS-232 Wiring

a) If you build the serial cable, you can use a 3-wire shield cable with shield connected to BOTH ends. Do not use the shield as GND. The ground wire (pin 5 of Sub-D 9) must be included inside the shield, like the RxD and TxD signals

b) Do not rely on an earthed PC to provide the TMC-3D earth connection! The TMC-3D must be earthed through a separate circuit. Most communication problems are caused by the lack of such connection

c) Always power-off all the IDS supplies before inserting/removing the RS-232 serial connector.

CAUTION! ***DO NOT CONNECT/DISCONNECT THE RS-232 CABLE WHILE THE TMC-3D IS POWERED ON. THIS OPERATION CAN DAMAGE THE TMC-3D***

3.2.7. CAN Communication – J10 Connector

Pin	Name	Type	Function
2	CAN_L	I/O	CAN-Bus negative line (negative during dominant bit)
3,6	CAN_GND	-	Reference ground for LO, HI and CAN_V+ signals
5	SHLD	-	Shield; Connected to frame
7	CAN_H	I/O	CAN-Bus positive line (positive during dominant bit)
1,4,8	n.c.	-	Not connected
9	CAN_V+	I	+24 V_{DC} isolated supply input

CAUTION! ***THE CANBUS CONNECTOR SIGNALS ARE ELECTRO-STATICALLY SENSITIVE AND SHALL BE HANDLED ONLY IN AN ESD PROTECTED ENVIRONMENT.***

Remarks:

a) The CAN network requires two 120Ω termination resistors even for short cables. These resistors are not included on the TMC-3D.

b) All 4 CAN signals are fully insulated from all other TMC-3D circuits (system ground – GND, IO ground – 0VPLC and Earth). Therefore, the CAN network requires a separate supply

3.2.7.1 Recommendations for CAN Wiring

a) Build CAN network using cables with 2-pairs of twisted wires (2 wires/pair) as follows: one pair for CAN_H with CAN_L and the other pair for CAN_V+ with CAN_GND. The cable impedance must be 105 ... 135 ohms (120 ohms typical) and a capacitance below 30pF/meter.

b) When total CAN bus length is below 5 meters, it is possible to use a standard phone straight-through cable (with parallel wires)

c) When total CAN bus length is over 40 meters, it is mandatory to use shielded twisted cables. Connect the cable shield to J10 pin 5 (SHLD)

d) Whenever possible, use daisy-chain links between the CAN nodes. Avoid using stubs. A stub is a "T" connection, where a derivation is taken from the main bus. When stubs can't be avoided keep them as short as possible. For 1 Mbit/s (worst case), the maximum stub length must be below 0.3 meters.

e) The 120Ω termination resistors must be rated at 0.2W minimum. Do not use winded resistors, which are inductive.

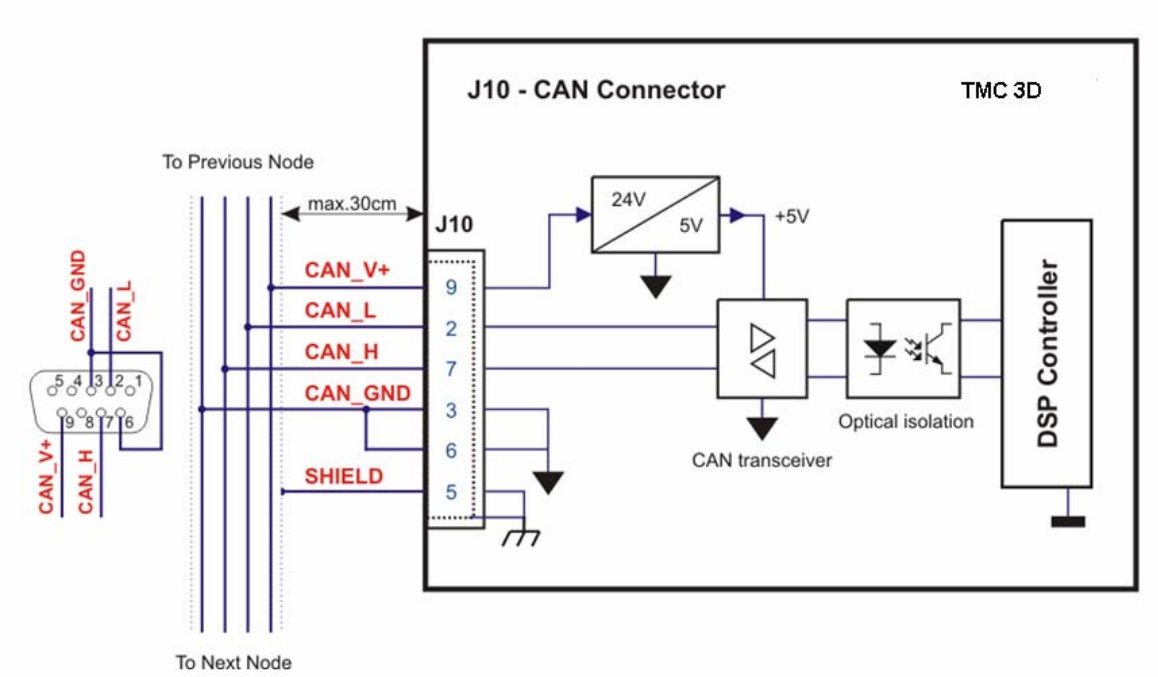


Figure 3.19. *J10 – CAN Connector*

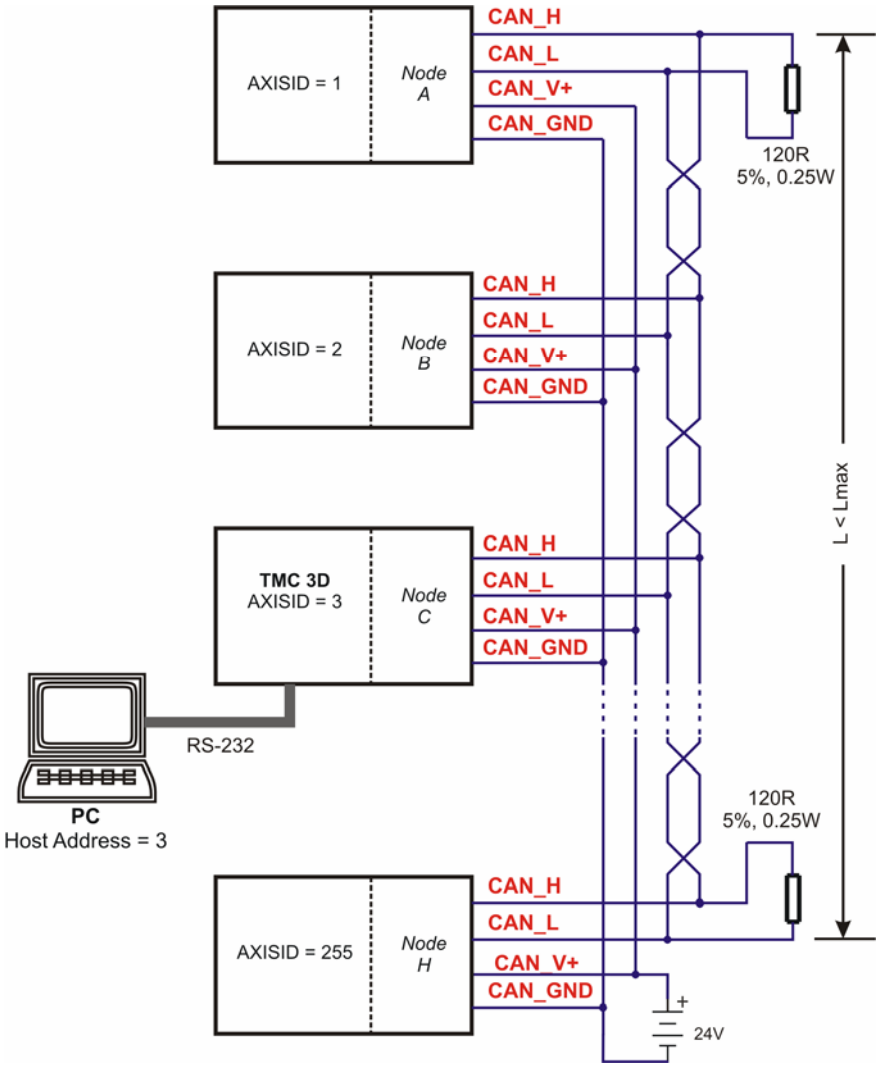


Figure 3.20. *Multiple-Axis CAN network*

3.2.8. Connectors Type and Mating Connectors

Connector	Function	Mating connector
J2	Motor & supply	Phoenix Contact MC 1.5/8-STF-3.5 [1]
J4	Serial	generic 9-pin Sub-D male
J10	CAN	generic 9-pin Sub-D female
J13	Feedback	generic 15-pin High Density Sub-D male
J9	Analog & 24 V digital I/O	generic 26-pin High Density Sub-D male

[1]. The mating connector accepts wires of 0.14 … 1.5 mm^2 (AWG35 … AWG16)

3.3. DIP-Switch Settings

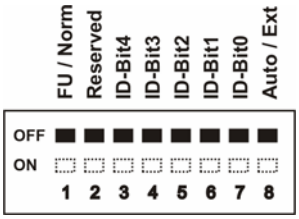


Figure 3.21. *SW1 – DIP Switch*

- **Position 1**: FU / Norm
  - ON: Enable **F**irmware **U**pdate
  - OFF: **Norm**al operation
- **Position 2**: Reserved
- **Positions 3 … 7**: ID-Bitx.
  Axis ID switches The drive axis/address number is set according with Table 3.1
- **Position 8**: Auto / Ext
  - ON: Sets the drive in AUTORUN mode. After power-on, the drive automatically executes a TML program from its internal E^2ROM.
  - OFF: Sets the drive in External (slave) mode. After power-on, the drive waits for commands from an external device.

Remark: All switches are sampled at power-up, and the drive is configured accordingly

Table 3.1. *Axis ID / Address configuration*

DIP Switch position					Axis ID
3	**4**	**5**	**6**	**7**	
ID – Bit4	**ID – Bit3**	**ID – Bit2**	**ID – Bit1**	**ID – Bit0**	
OFF	OFF	OFF	OFF	OFF	255
OFF	OFF	OFF	OFF	ON	1
OFF	OFF	OFF	ON	OFF	2
OFF	OFF	OFF	ON	ON	3
OFF	OFF	ON	OFF	OFF	4
OFF	OFF	ON	OFF	ON	5
OFF	OFF	ON	ON	OFF	6
OFF	OFF	ON	ON	ON	7
OFF	ON	OFF	OFF	OFF	8
OFF	ON	OFF	OFF	ON	9
OFF	ON	OFF	ON	OFF	10
OFF	ON	OFF	ON	ON	11
OFF	ON	ON	OFF	OFF	12
OFF	ON	ON	OFF	ON	13
OFF	ON	ON	ON	OFF	14
OFF	ON	ON	ON	ON	15
ON	OFF	OFF	OFF	OFF	16
ON	OFF	OFF	OFF	ON	17
ON	OFF	OFF	ON	OFF	18
ON	OFF	OFF	ON	ON	19
ON	OFF	ON	OFF	OFF	20
ON	OFF	ON	OFF	ON	21
ON	OFF	ON	ON	OFF	22
ON	OFF	ON	ON	ON	23
ON	ON	OFF	OFF	OFF	24
ON	ON	OFF	OFF	ON	25
ON	ON	OFF	ON	OFF	26
ON	ON	OFF	ON	ON	27
ON	ON	ON	OFF	OFF	28
ON	ON	ON	OFF	ON	29
ON	ON	ON	ON	OFF	30
ON	ON	ON	ON	ON	31

Technosoft motion controllers and drives/motors can be set with axis ID values from 1 to 255.

3.4. LED Indicators

LED Color	Function
Green	Lit after power-on when the drive initialization ends. Turned off when an error occurs
Red	Turned on when the power stage error signal is generated or when OUT4 is set low

3.5. First Power-Up

In order to setup the TMC-3D for your application you need to communicate with it. The easiest way is via an RS-232 serial link between your PC and the TMC-3D. Therefore, before the first power-up, check the following:

- Power supply connections and their voltage levels
- Motor connections
- Serial cable connections
- DIP switch positions: all shall be OFF (not pressed).
- EasyMotion Studio is installed on the PC which is serially connected with the drive (see chapter Step 2. Motion Controller Setup).

4. Step 2. Motion Controller Setup

4.1. Installing EasyMotion Studio

EasyMotion Studio is a PC software platform for the setup and motion programming of the Technosoft motion controllers/drives. EasyMotion Studio includes **EasySetUp** for the drive/motor setup, and a **Motion Wizard** for the motion programming. The Motion Wizard provides a simple, graphical way of creating motion programs written in Technosoft Motion Language (TML).

A demo version of EasyMotion Studio can be downloaded ***free of charge*** from Technosoft web page. EasyMotion Studio comes with an ***Update via Internet tool*** through which you can check if your software version is up-to-date, and when necessary download and install the latest updates. EasyMotion Studio package includes a firmware programmer through which you can update your drive firmware to the latest revision.

On request, EasyMotion Studio can be provided on a CD too. In this case, after installation, use the update via internet tool to check for the latest updates. Once you have started the installation package, follow its indications.

The demo version of EasyMotion Studio can be transformed into a fully functional version by introducing the license serial number in the menu command **Help | Enter registration info…**.

4.2. Getting Started with EasyMotion Studio

Using EasyMotion Studio you can quickly configure the multi-axis system, setup each slave axis and program the motion sequences for your application.

The TMC-3D can be:

- directly connected with your PC via a serial RS 232 link
- part of a CANbus network where the PC is serially linked with one of the slave drives.

EasyMotion Studio works with **projects**. A project can contain one or several **motion controller applications** and several **slave applications**.

The motion controller application contains the description of the multi-axis system – the multi-axis configuration data, the setup configuration for the embeded drive and the motion sequences which implement the functionality of the multi-axis application. The motion controller application has 3 components: **Axis Selection** – describes the multi-axis system, **Setup** – embeded drive configuration and Motion program.

The slave applications describe the setup one axis. It has 2 components: the **Setup** data and the **Motion** program and an associated axis number: an integer value between 1 and 255.

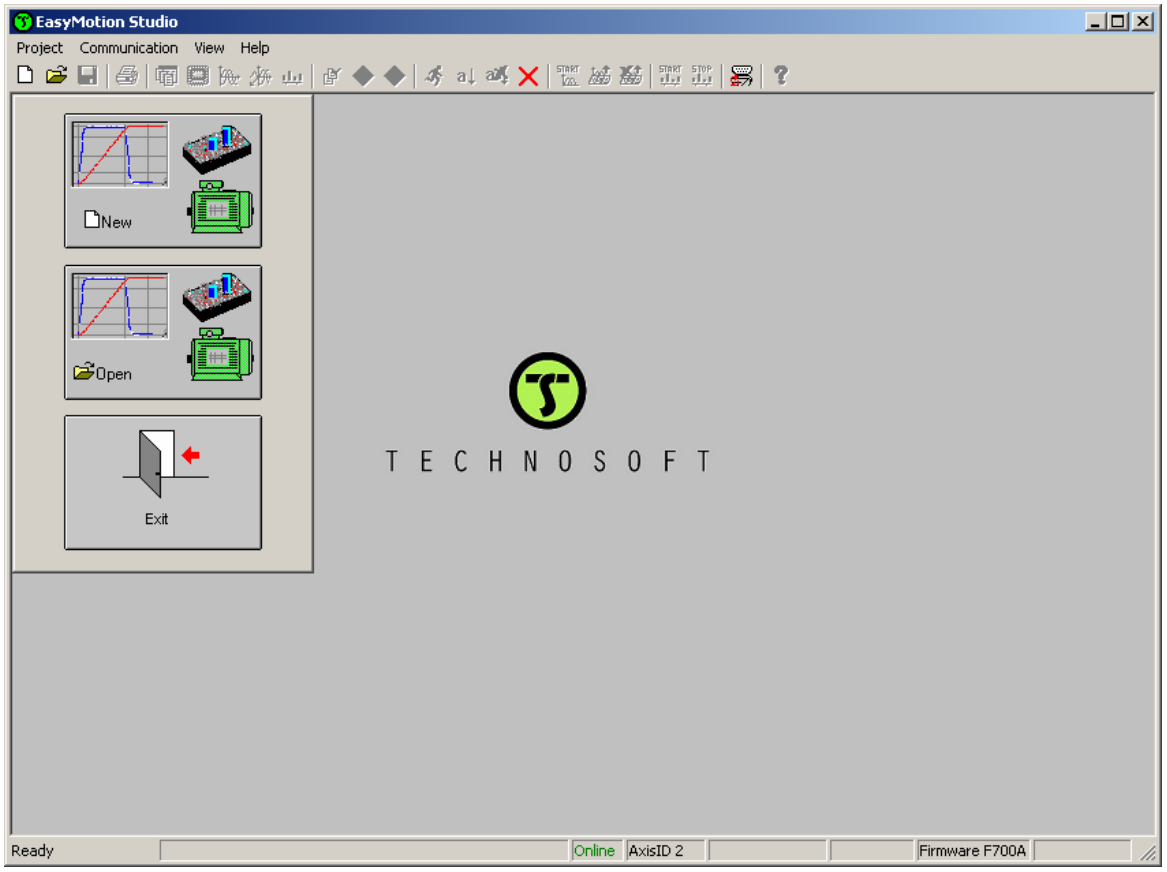
An application may be used either to describe:

1. One axis in a multi-axis system
2. An alternate configuration (set of parameters) for the same axis.

In the first case, each application has a different axis number corresponding to the axis ID of the drives/motors from the network. All data exchanges are done with the drive/motor having the same address as the selected application. In the second case, all the applications have the same axis number.

4.3. Establish communication

EasyMotion Studio starts with an empty window from where you can create a **New** project or **Open** a previously created project which was saved on your PC. When you start a new project, EasyMotion Studio automatically creates a first application. Additional applications for the slave axes can be added later.

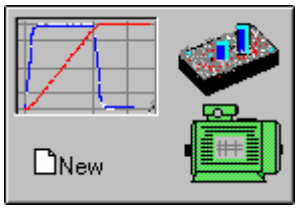


Before selecting one of the above options, you need to establish the communication with the motion controller or the drive you want to commission. Use menu command **Communication | Setup** to check/change your PC communication settings. Press the **Help** button of the dialogue opened. Here you can find detailed information about how to setup the motion controller and do the connections. Power on the drive, then close the Communication | Setup dialogue with OK. If the communication is established, EasyMotion Studio displays in the status bar (the bottom line)

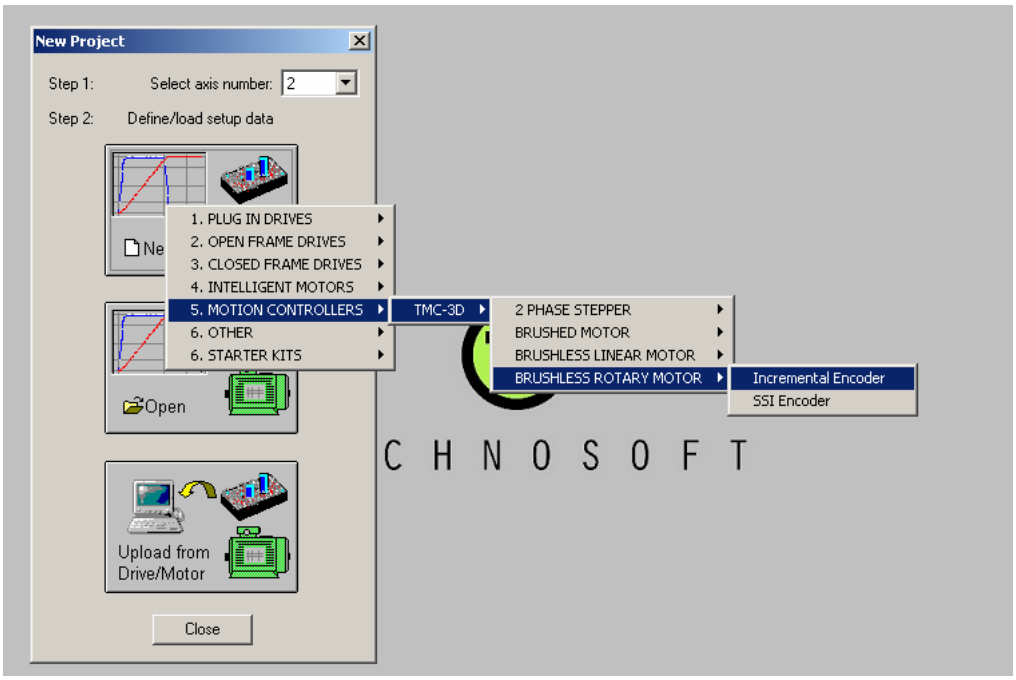
the text "**Online**" plus the axis ID of your motion controller/drive and its firmware version. Otherwise the text displayed is "**Offline**" and a communication error message tells you the error type. In this case, return to the Communication | Setup dialogue, press the Help button and check troubleshoots

Remark: *When first started, EasyMotion Studio tries to communicate via RS-232 and COM1 with a drive having axis ID=255 (default communication settings). If your drive is powered with all the DIP switches OFF and it is connected to your PC port COM1 via an RS-232 cable, the communication shall establish automatically. If the drive has a different axis ID and you don't know it, select in the Communication | Setup dialog at "Axis ID of drive/motor connected to PC" the option* ***Autodetected****.*

4.4. Motion Controller Application

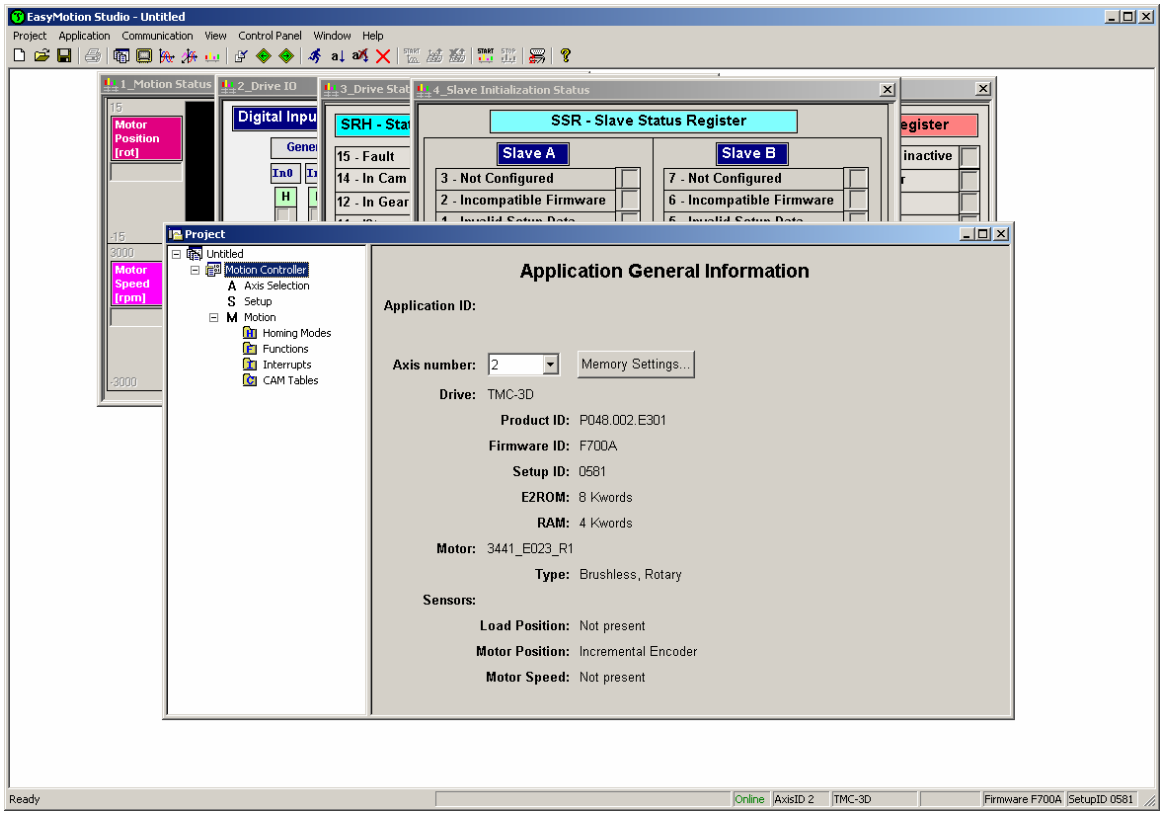
Press **New** button 

to open the "New Project" dialogue. Set the axis number for motion controller application equal with your motion controller Axis ID. The initial value proposed is 255 which is the default axis ID of the motion controllers and drives/motors. Press **New** button and select the product type (Motion Controllers) followed by product selection.



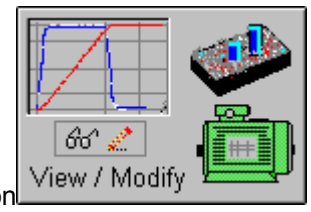
The selection continues with the motor technology (for example: brushless or brushed) and type of feedback device (for example: Incremental encoder, SSI encoder).

EasyMotion Studio opens the **Project** window where on the left side you can see the structure of a project. At beginning the new project is named “Untitled” and the first application is named Motion Controller. The Motion Controller application has 3 components: **A** Axis Selection, **S** Setup and **M** Motion (program).

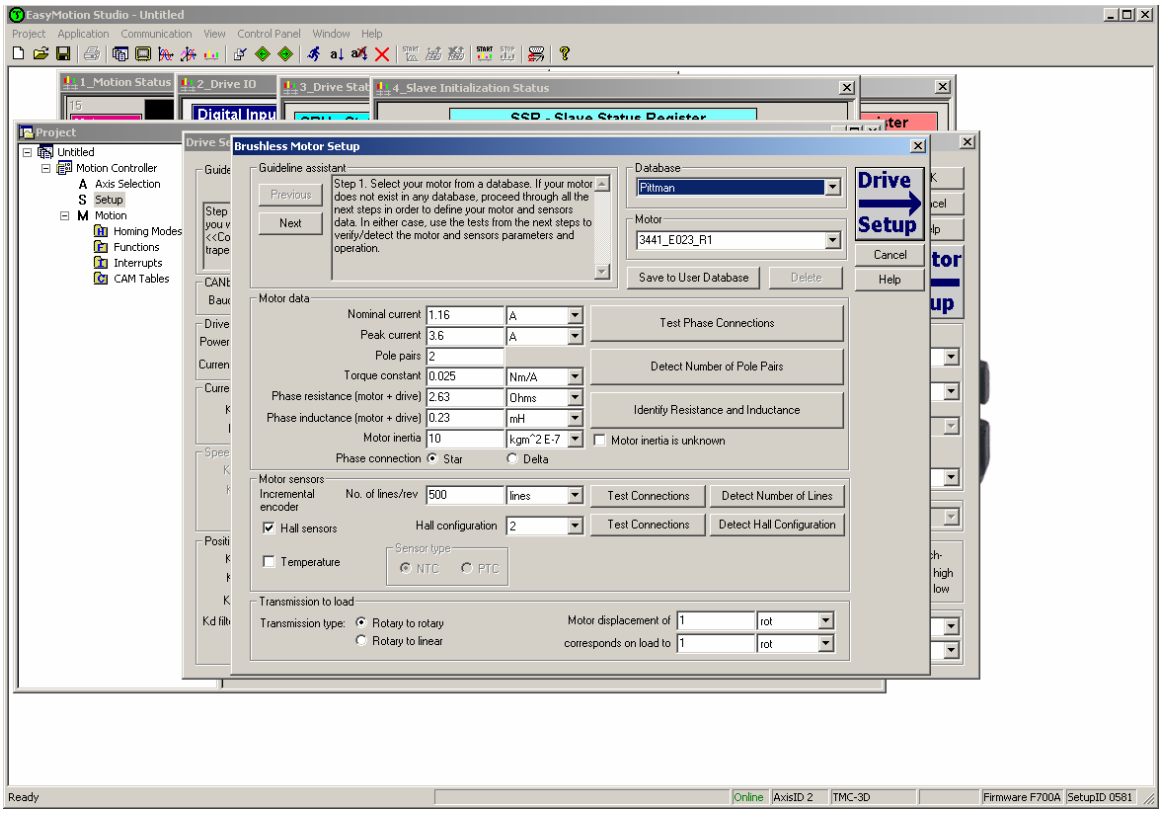


4.5. Embeded Drive Setup

From the project tree select **S Setup** to access the setup data for drive part of TMC-3D.

Press the View/Modify button 

to open the setup dialogs: **Motor Setup** and **Drive Setup** through which you can configure and parameterize the drive part of Technosoft motion controller.



In the **Motor Setup** dialog you can introduce the data of your motor and the associated sensors. Data introduction is accompanied by a series of tests having as goal to check the connections to the drive and/or to determine or validate a part of the motor and sensors parameters. In the **Drive Setup** dialog you can configure and parameterize the drive part of the motion controller for your application. In each dialog you will find a **Guideline Assistant**, which will guide you through the whole process of introducing and/or checking your data. Close the Drive Setup dialog with **OK** to keep all the changes regarding the motor and the drive setup.

4.6. Changing the Axis ID of the Motion Controller

The axis ID of an TMC-3D drive can be set in 2 ways:

- Hardware (H/W) – according with the DIP switch selection in the range 1 to 31 or 255 (see 3.3 DIP-Switch Settings)
- Software – any value between 1 and 255, stored in the setup table

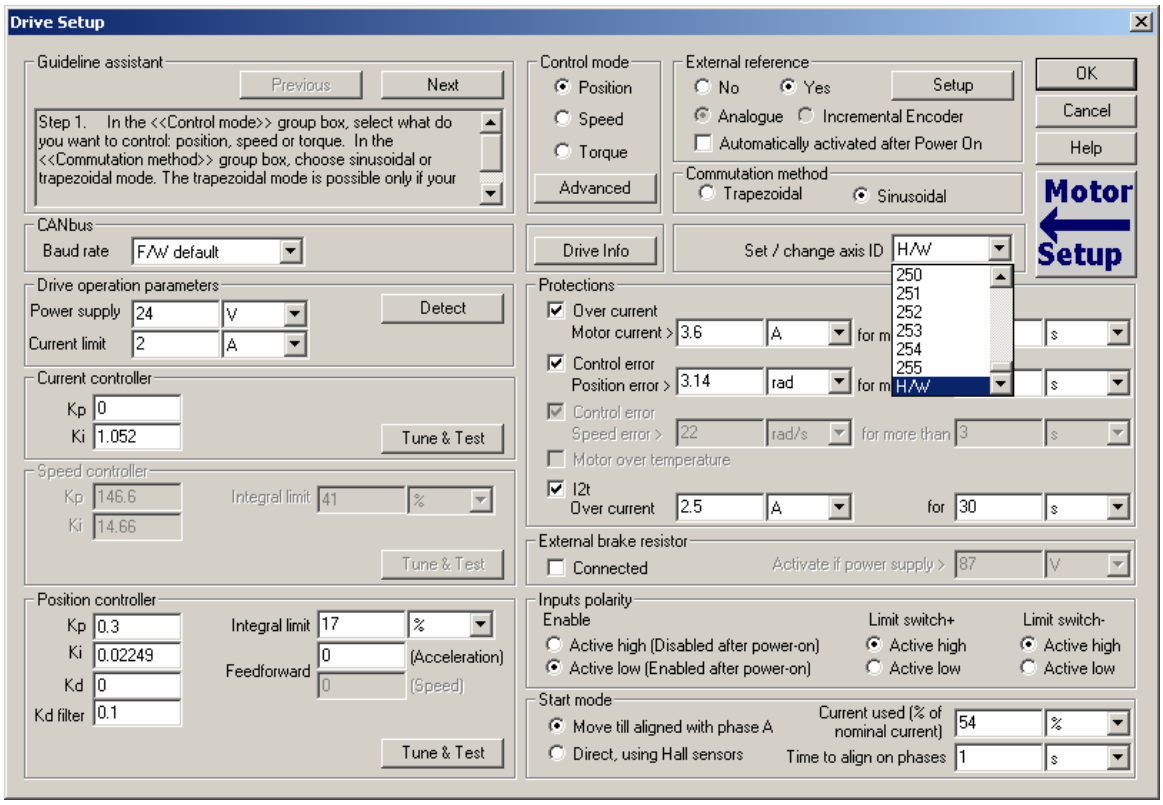
The axis ID is initialized at power on, using the following algorithm:

a) If a valid setup table exists, with the value is read from it. This value can be an axis number 1 to 255 or can indicate that axis ID will be set according with DIP switch selection

b) If the setup table is invalid, with the last value set with a valid setup table. This value can be an axis number 1 to 255 or can indicate that axis ID will be set according with DIP switch selection

c) If there is no axis ID set by a valid setup table, according with DIP switch selection

Remark: *If an Axis ID was previously set by software and its value is not anymore known, you can find it by selecting in the Communication | Setup dialogue at "Axis ID of drive/motor connected to PC" the option* ***Autodetected****. Apply this solution only if this drive is connected directly with your PC via an RS-232 link. If this drive is part of a CANbus network and the PC is serially connected with another drive, use the menu command* ***Communication | Scan Network****.*



4.7. Setting CANbus Rate

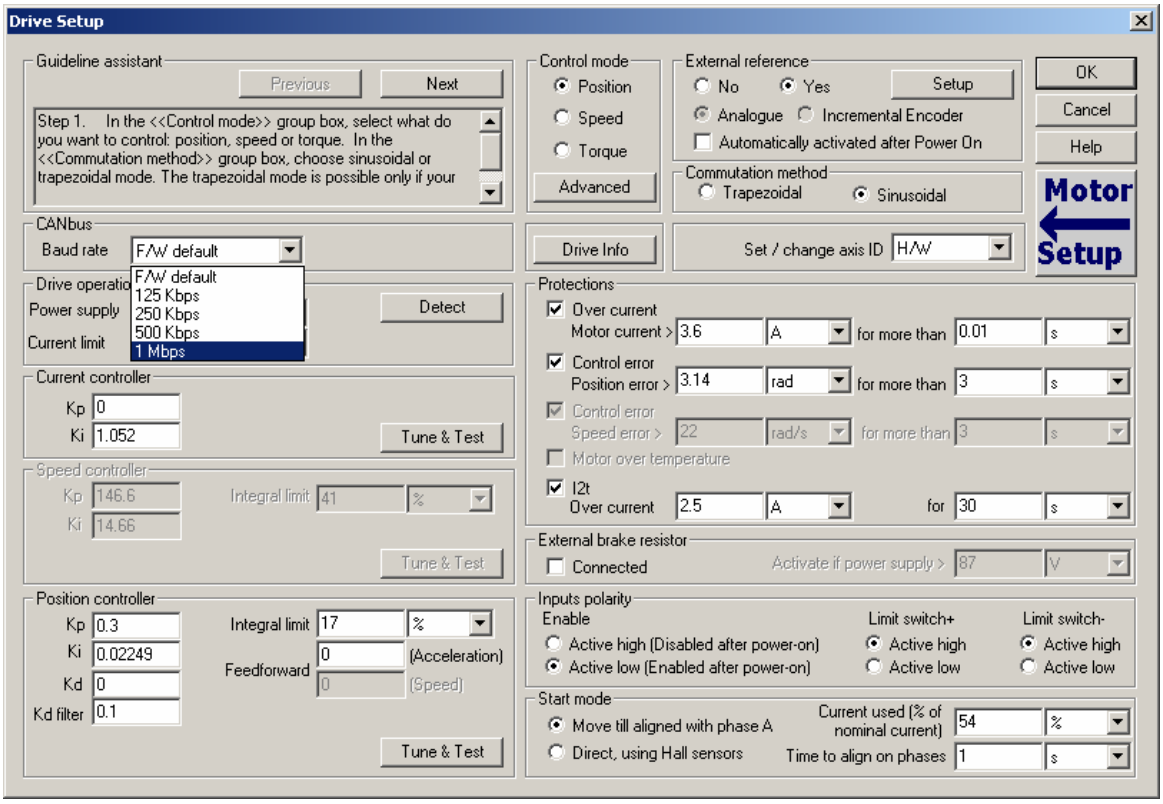
The TMC-3D can work with the following rates on the CAN: 125kHz, 250kHz, 500KHz, 1MHz. In the Drive Setup dialogue you can choose the initial CAN rate after power on. This information is stored in the setup table. The CAN rate is initialized using the following algorithm:

a) If a valid setup table exists, with the CAN rate value read from it. This can be any of the supported rates or can indicate to use the firmware default (F/W default) value, which is 500kHz

b) If the setup table is invalid, with the last CAN rate value set with a valid setup table. This can be any of the supported rates or can indicate to use the firmware default (F/W default) value

c) If there is no CAN rate value set by a valid setup table, with the firmware default value i.e. 500kHz

Remark: *The same CAN baud rate must be set for motion controller and slave axes*



4.8. Download Setup Data to EEPROM

Press the **Download to Drive/Motor** button to download your setup data in the motion controller EEPROM memory, in the *setup table*. From now on, at each power-on, the setup data is copied into the motion controller RAM memory which is used during runtime.

Remark: *The setup data becomes active after drive reset or when the power supply is cycled.*

4.9. Evaluate the behaviour of motion controller drive part (optional)

You can use the **Data Logger** or the **Control Panel** evaluation tools to quickly measure and analyze your application behavior. In case of errors like protections triggered, use the Drive Status control panel to find the cause.

4.10. Creating an Image File with the Setup Data

Once you have validated the embeded drive setup, you can use the **Application | Create EEPROM Programmer File | Setup…** menu command to generate a software file (with extension **.sw**) with all the setup data. The software file provides a convenient way to automate the programming of TMC-3D.

The software file is a text file that can be read with any text editor. It contains blocks of data separated by an empty raw. Each block of data starts with the block start address, followed by data values to place in ascending order at consecutive addresses: first data – to write at start address, second data – to write at start address + 1, etc. All the data are hexadecimal 16- bit values (maximum 4 hexadecimal digits). Each raw contains a single data value. When less then 4 hexadecimal digits are shown, the value must be right justified. For example 92 represent 0x0092.

The **.sw** file can be programmed into a drive:

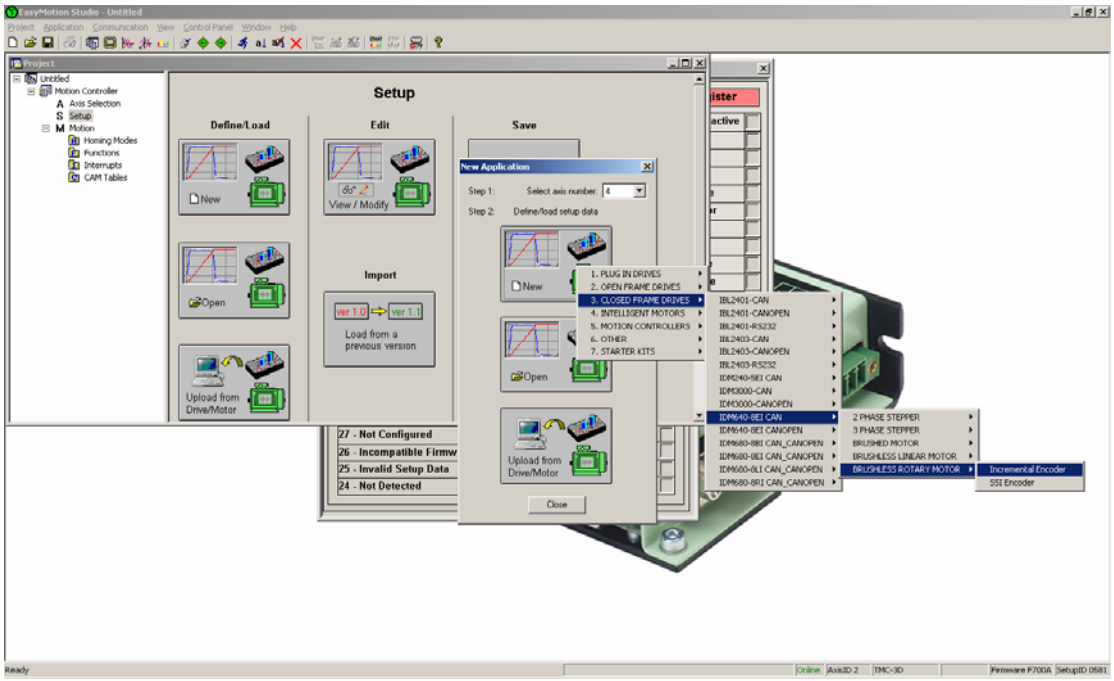
- from a host PC or PLC, using the TML_LIB functions for writing data into the drive EEPROM
- using the EEPROM Programmer tool, which comes with EasyMotion Studio but may also be installed separately. The EEPROM Programmer was specifically designed for repetitive fast and easy programming of **.sw** files into the Technosoft drives during production.

5. Step 3. Slave Axes Setup

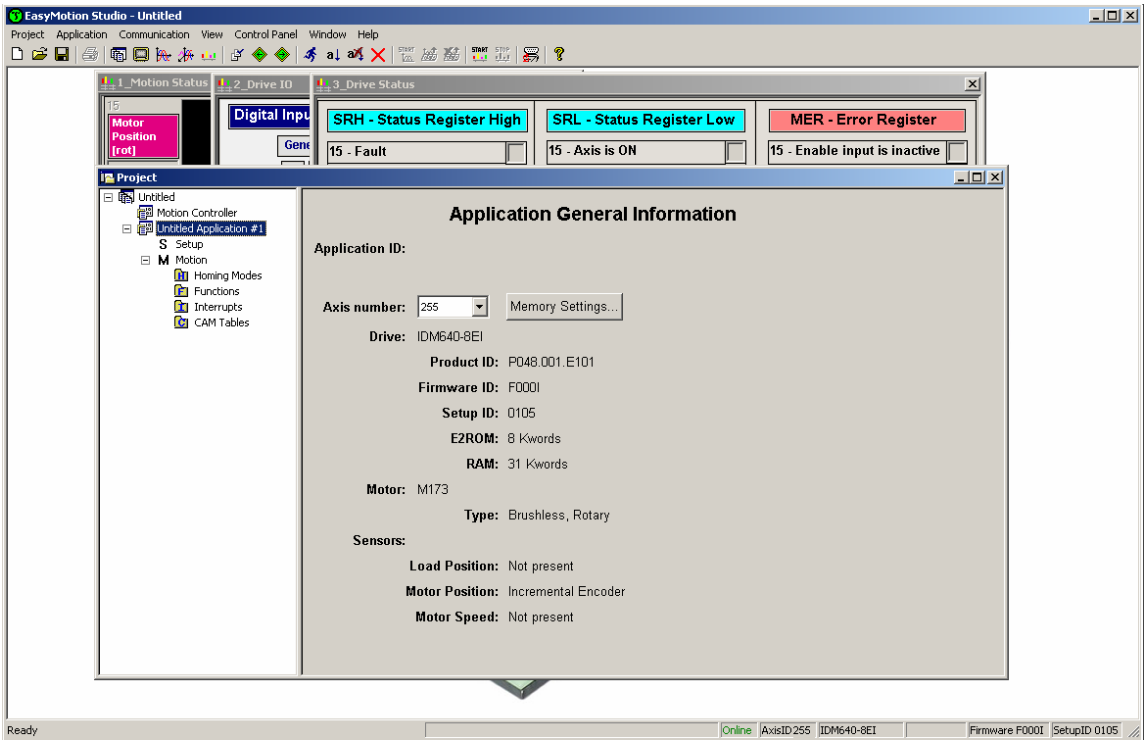
Each axis from the multi-axis system requires a separate application with Setup and Motion information. Use the **Application | New…** command to add the slave applications in motion controller project.

Set the Axis ID for the slave equal with your intelligent drive/motor Axis ID. Press **New** button and select the product type followed by product, motor technology and feedback type.

Remark: *Each CAN network node must have a unique Axis ID. The mechanism of setting the Axis ID for the intelligent drives is identical with the one used for TMC-3D (section 4.6 Changing the Axis ID of the Motion Controller)*



The commissioning of slave drives/motors implies the same procedure as for the embeded drive in motion controller: motor and drive configuration through **Motor Setup** and **Drive Setup** dialogs (section 4.5 Embeded Drive Setup), **Axis ID configuration** (section 4.6 Changing the Axis ID of the Motion Controller), **download setup** data in EEPROM (section 4.8 Download Setup Data to EEPROM).



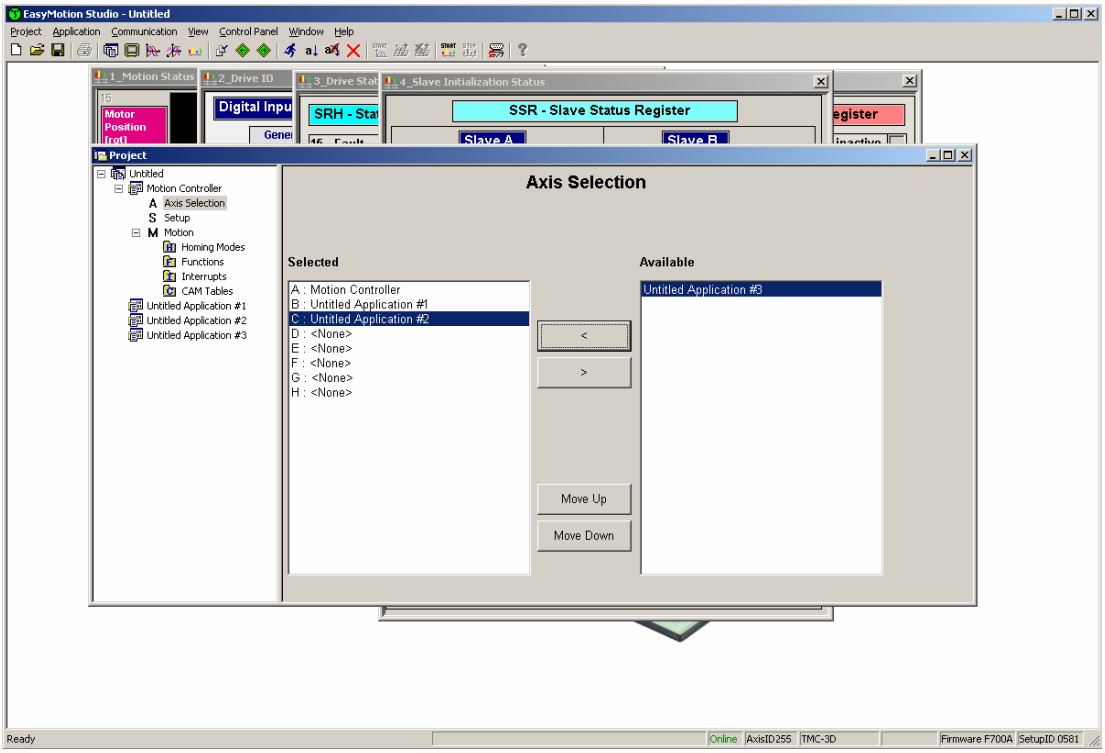
The slaves' setup data can be stored in individual software files, similar with the one generated for motion controller (section 4.10 Creating an Image File with the Setup Data), allowing easy programming of the drives/motor, outside of EasyMotion Studio environment.

6. Step 4. Motion Programming

6.1. Multi-axis configuration

When the commissioning for all the slave applications is finished the multi-axis application can be configured. The multi-axis configuration is done from the Motion Controller Application.

Back in the Motion Controller application go to the "**A** Axis Selection" component. Inside the "**Available**" section you will find the Slaves Applications. Notice that the Motion Controller Application is also listed since the motion controller itself can be one of the slaves. From all the available applications a maximum number of eight can be configured as slaves of the motion controller. A drive/motor is consider to be a slave when the associated application is moved to the "**Selected**" field. You can use the arrows to move the applications between Available field and Selected field.



The slaves of the motion controller are symbolically named **A**, **B**, **C**, **D**, **E**, **F**, **G**, **H**. Once the an application is moved to the "Selected" field, it will be addressed with his corresponding alias.

When an application is selected as one of the slaves, the motion controller will modify its group ID according to the slave position (i.e. Slave A = group ID 1, Slave B = group ID 2, .., Slave H = group ID 8). The motion controller will exclusively use group messages to send commands to the slaves.

Remark: *By default, all Technosoft drives/motors belong to group 1. If the multi-axis machine requires independent axes, i.e. not controlled by TMC-3D, then the user must reconfigure them to become part of group 0 or other available group. The group information can be changed with TML instructions GROUPID or REMGRID. The GROUPID and REMGRID can be executed from TML application or sent by a host device.*

Example 1: *Network configuration: 6 Technosoft drives, 4 drives are configured in a Motion Controller application (Slaves A, B, C, D) and 2 drives are independent axes.*

The 2 stand alone drives need to be configured to one of the group IDs 5, 6, 7 or 8. Using the (example GROUPID (7)). Not doing this would mean that they will receive the same messages as Slave A.

Example 2: *Network configuration: 10 Technosoft drives, 8 drives are configured in a Motion Controller application (Slaves A, B, C, D, E, F, G, H) and 2 drives are independent axes.*

Because in this configuration all the group IDs are occupied with the axes controlled by the motion controller then the independent axes must be configured to belong to group 0 with the REMGRID command (example REMGRID(1, 2, 3, 4, 5, 6, 7, 8) remove all groups).

6.2. Generate coordinated multi-axis profiles

The TMC-3D motion controller can coordinate up to 8 axes (including itself) connected together in a CAN network. Taking advantage of the intelligence embedded in all the Technosoft drives, the TMC-3D can either generate complex 2D/3D trajectories or coordinate various single axis trajectories inside a distributed controlled network. It employs an efficient synchronization mechanism that insures the internal clock of each drive remains synchronous to the Motion Controller. In case of the 2D/3D trajectories instead of sending the position information at each position control loop, the TMC-3D sends PVT (**P**osition, **V**elocity, **T**ime) points at predefined time intervals, minimizing the traffic on the CAN network. Based on the received PVT points, the slaves reconstruct the desired trajectory performing a 3rd order interpolation between the points.

6.2.1. Coordinated multi axis modes and application management

The 2D/3D trajectories can be generated using the Linear Interpolation and/or the Vector Mode wizards.

- **Linear Interpolation**

Linear Interpolation allows the generation of complex 2D or 3D paths using linear segments. The master generates the reference for each projection and sends the position commands as PVT (position, velocity and time) points.

- **Vector Mode**

Vector Mode allows the generation of 2D profiles that need to combine linear segments with circular ones. It also has the possibility to control a third axis which will remain tangent to the trajectory. This is useful for example in cutting applications. The master generates the reference for each projection and sends the position commands as PVT (position, velocity and time) points.

- **Error handling**

The motion controller has a powerful mechanism of detecting and handling the errors that may occur in the system:

- The TMC3D can program the slaves to report error situations. The user can configure, at initialization time, which errors will be reported by the slaves.
- The TMC-3D can monitor the slaves presence in the CAN network through the **Slave Guarding** functionality. When the Slave Guarding is activated the motion controller periodically interrogates the slaves drives/motors and monitors their answers. A node failure error is triggered if a slave fails to respond back to the master within one slave guarding period. The Slave Guarding is configurable by the user

In case of a slave error or node failure the TMC-3D can trigger a TML interrupt and execute application specific TML code. The TML interrupt can be enabled/disabled by the user.

- **Program flow control**

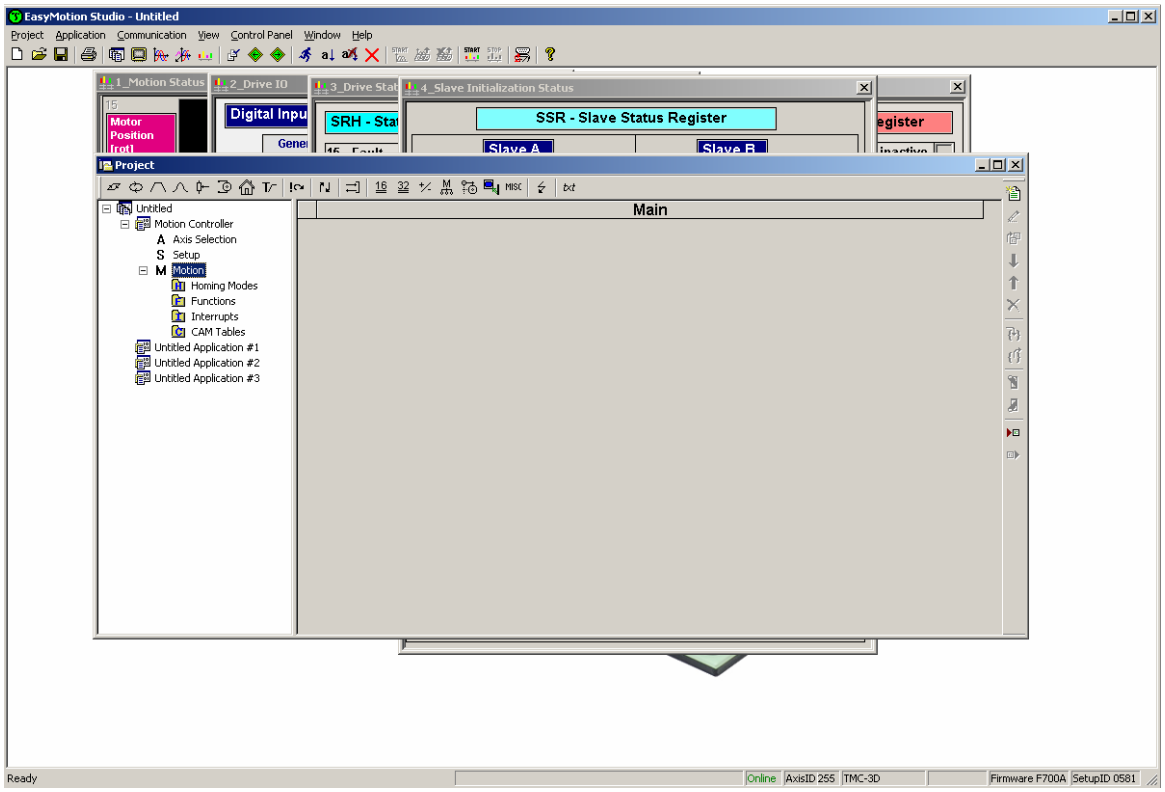
A new type of multi axis events was added in order to maintain a high level of synchronization between the master and the slaves in the multi-axis application.

- **Machine setup**

The TMC-3D can command different homing procedures on each slave in order to find the machine home (start) position. The homing procedures are TML function with application specific TML code which can be different for each slave. The routines are created for each slave application and downloaded in the EEPROM memory of the slave during commissioning. During operation the TMC-3D commands the execution of these routines.

6.2.1.1 Motion programming

In the project window left side, select "**M** Motion", for motion programming. This automatically activates the **Motion Wizard**.



The Motion Wizard offers you the possibility to program all the motion sequences using high level graphical dialogues which automatically generate TML code. When activated, Motion Wizard adds a set of toolbar buttons in the project window just below the title. The Motion Wizard provides a fast way to develop complex TML programs. Each button opens a dialog where the user can set specific parameters. Based on these parameters the Wizard will generate the TML code. The generated TML instructions are not a simple text included in a file, but a motion object. Therefore the output of the Motion Wizard is a collection of motion objects.

The major advantage of encapsulating programming instructions in motion objects is that you can very easily manipulate them. For example, you can:

- Save and reuse a complete motion program or parts of it in other applications
- Add, delete, move, copy, insert, enable or disable one or more motion objects
- Group several motion objects and work with bigger objects that perform more complex functions.

After the TML code has been generated, the user can test it by running the application; press the **Run** button. At this point the following operations are performed:

- A TML program is created by inserting the motion objects into a predefined template
- The TML program is compiled and downloaded to the TMC-3D
- The TML program execution is started

For learning how to send TML commands from your host/master, using one of the communication channels and protocols supported by the drives use menu command **Application | Binary Code Viewer…** Using this tool, you can get the exact contents of the messages to send and of those expected to be received as answers.

6.2.1.2 Evaluate motion application performances

EasyMotion Studio includes a set of evaluation tools like the **Data Logger**, the **Control Panel** and the **Command Interpreter** which help you to quickly measure and analyze your motion application.

6.2.2. Creating an Image File with the Setup Data and the TML Program

Once you have validated your application, you can create with the menu command **Application | Create EEPROM Programmer File** a software file (with extension **.sw**) which contains all the data to write in the EEPROM of your drive. This includes both the setup data and the motion program. For details regarding the **.sw** file format and how it can be programmed into a drive, see paragraph **Error! Reference source not found.**

6.3. Advanced features

The TMC-3D offers many advance management features for the multi axis application. All features presented below require usage of EasyMotion Studio as TML programming tool.

6.3.1. Error handling

The TMC-3D has a powerful and highly customizable error handling mechanism. Each slave can be programmed from the Motion Controller application in order to report in case of a specific error. All the slave applications have their own mechanism of reacting in case of an error (i.e. short circuit, under voltage, control error etc.), but this is not enough. In a multi axis application the motion controller needs to take proper action for all the nodes (even they don not have any errors) in case one node fails. The Motion Controller can program each slave to report what errors can affect the good functioning of the machine. For example in a 2D cutting machine if the X axis for some reason can’t follow the commanded trajectory anymore then the Motion Controller

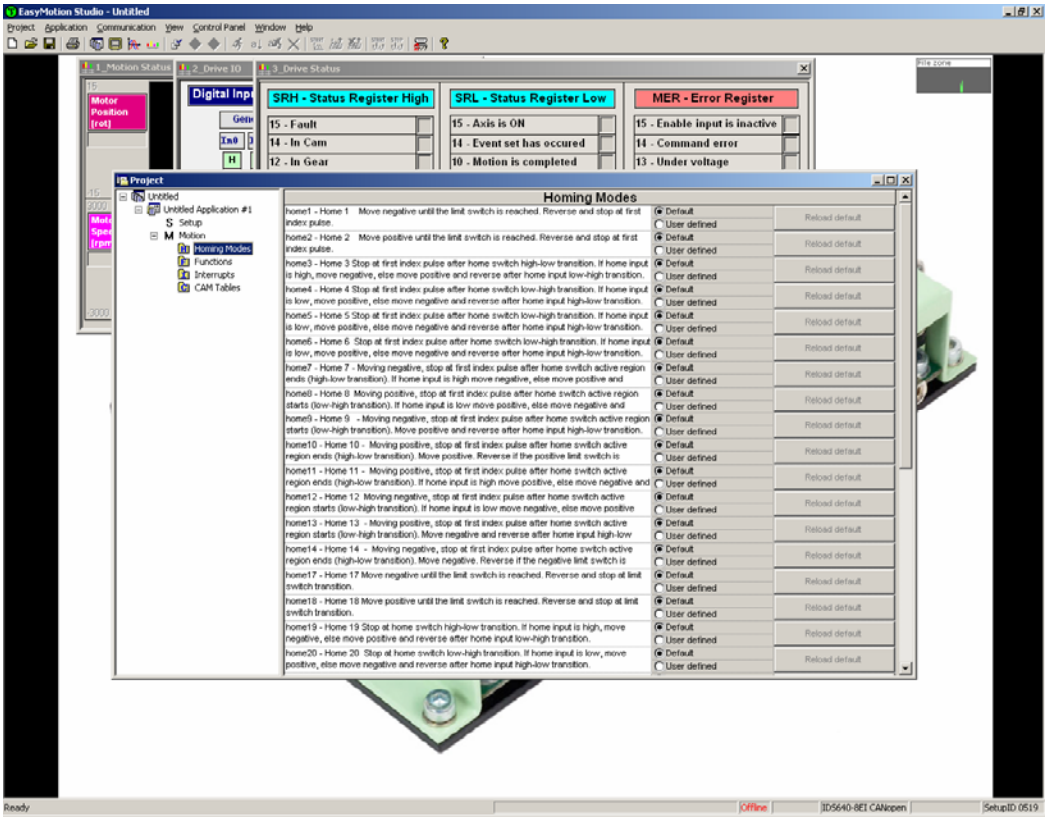
must stop the Y axis also. In order for him to do so, it must configure the X and the Y axes to report if a Control Error occurs.

6.3.2. Using TML Functions to Split Motion between Motion Controller and slaves axes

With Technosoft intelligent drives you can really distribute the intelligence between the Motion Controller and the slave drives in complex multi-axis applications. Instead of trying to command each step of an axis movement, you can program the drives using TML to execute complex tasks and inform the Motion Controller when these are done. Thus the Motion Controller task may be reduced to call TML functions (with possibility to abort their execution) stored in the drives EEPROM and wait for drive to report the finalization of the TML functions.

6.3.3. Customizing the Homing Procedures

All Technosoft motion controllers/drives have Homing procedures and if needed, any of these homing modes can be customized. In order to do this you need to select the Homing Modes from your application tree and in the right side choose “User defined” for one of the Homing procedures. Following this operation the selected procedure will be listed under Homing Modes in a sub-tree, with the name *HomeX* where X is the number of the selected homing.



If you select the *HomeX* procedure, on the right side you'll see the TML code implementing it. The homing routine can be customized according to your application needs. It's calling name and method remain unchanged. Further, the TMC-3D can command the execution of a specific homing for every application. In this way for example the TMC-3D can put the machine in the starting position.

6.3.4. Customizing the Drive Reaction to Fault Conditions

Similarly to the homing modes, the default service routines for the TML interrupts can be customized according to your application needs. However, as most of these routines handle the drive reaction to fault conditions, it is mandatory to keep the existent functionality while adding your application needs, in order to preserve the correct protection level of the drive. The procedure for modifying the TML interrupts is similar with that for the homing modes.

6.4. Using Motion Libraries for PC-based Systems

A **TML Library for PC** is a collection of high-level functions allowing you to control from a PC a network of Technosoft intelligent drives. It is an ideal tool for quick implementation on PCs of motion control applications with Technosoft products.

With the TML Motion Library functions you can: communicate with a motion controller/drive via any of its supported channels (RS-232, CAN-bus), send motion commands, get automatically or on request information about motion controller/drive status, check and modify its setup parameters, read inputs and set outputs, etc.

The TML Motion Library can work under a **Windows** or **Linux** operating system. Implemented as a .dll/.so, it can be included in an application developed in **C/C++, C#, Visual Basic, Delphi Pascal** or **Labview**.

Using a TML Motion Library for PC, you can focus on the main aspects of your application, while the motion programming part can be reduced to calling the appropriate functions and getting the confirmation when the task was done.

7. Scaling Factors

Technosoft motion controllers/drives work with parameters and variables represented in the drive internal units (IU). These correspond to various signal types: position, speed, current, voltage, etc. Each type of signal has its own internal representation in IU and a specific scaling factor. This chapter presents the drive internal units and their relation with the international standard units (SI).

In order to easily identify them, each internal unit has been named after its associated signal. For example the **position units** are the internal units for position, the **speed units** are the internal units for speed, etc.

7.1. Position units

7.1.1. Brushless / DC brushed motor with quadrature encoder on motor

The internal position units are encoder counts. The correspondence with the load **position in SI units**[1] is:

For rotary motors: $$Load_Position[SI] = \frac{2 \times \pi}{4 \times No_encoder_lines \times Tr} \times Motor_Position[IU]$$

For linear motors: $$Load_Position[SI] = \frac{Encoder_accuracy}{Tr} \times Motor_Position[IU]$$

where:

No_encoder_lines – is the rotary encoder number of lines per revolution

Encoder_accuracy – is the linear encoder accuracy i.e. distance in [m] between 2 pulses

Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units

[1] SI units for position are: [rad] for a rotary movement, [m] for a linear movement

7.1.2. Brushless motor with absolute SSI encoder on motor

The internal position units are encoder counts. The motor is rotary. The correspondence with the load **position in SI units**[1] is:

$$\text{Load_Position[SI]} = \frac{2 \times \pi}{2^{\text{No_bits_resolution}} \times \text{Tr}} \times \text{Motor_Position[IU]}$$

where:

No_bits_resolution – is the SSI encoder resolution in bits per revolution

Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units

7.1.3. DC brushed motor with quadrature encoder on load and tacho on motor

The internal position units are encoder counts. The motor is rotary and the transmission is rotary-to-rotary. The correspondence with the load position in SI units is:

$$\text{Load_Position[rad]} = \frac{2 \times \pi}{4 \times \text{No_encoder_lines}} \times \text{Load_Position[IU]}$$

where:

No_encoder_lines – is the encoder number of lines per revolution

7.1.4. DC brushed motor with absolute SSI encoder on load and tacho on motor

The internal position units are encoder counts. The motor is rotary and the transmission is rotary-to-rotary. The correspondence with the load position in SI units is:

$$\text{Load_Position[SI]} = \frac{2 \times \pi}{2^{\text{No_bits_resolution}}} \times \text{Load_Position[IU]}$$

where:

No_bits_resolution – is the SSI encoder resolution in bits per revolution

[1] SI units for position are: [rad] for a rotary movement, [m] for a linear movement

7.1.5. Stepper motor open-loop control. No feedback device

The internal position units are motor μsteps. The correspondence with the load **position in SI units** is:

$$\text{Load_Position[SI]} = \frac{2 \times \pi}{\text{No_}\mu\text{steps} \times \text{No_steps} \times \text{Tr}} \times \text{Motor_Position[IU]}$$

where:

No_steps – is the number of motor steps per revolution

No_μsteps – is the number of micro-steps per step. You can read/change this value in the "Drive Setup" dialogue from EasyMotion Studio.

Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units

7.1.6. Stepper motor closed-loop control. Incremental encoder on motor

The internal position units are motor encoder counts. The correspondence with the load **position in SI units** [1] is:

$$\text{Load_Position[SI]} = \frac{2 \times \pi}{4 \times \text{No_encoder_lines} \times \text{Tr}} \times \text{Motor_Position[IU]}$$

where:

No_encoder_lines – is the motor encoder number of lines per revolution

Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units

7.1.7. Stepper motor open-loop control. Incremental encoder on load

The internal position units are load encoder counts. The transmission is rotary-to-rotary. The correspondence with the load position in SI units is:

$$\text{Load_Position[SI]} = \frac{2 \times \pi}{4 \times \text{No_encoder_lines}} \times \text{Load_Position[IU]}$$

where:

No_encoder_lines – is the rotary encoder number of lines per revolution

Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units

[1] SI units for position are [rad] for a rotary movement , [m] for a linear movement

7.2. Speed units

The internal speed units are internal position units / (slow loop sampling period) i.e. the position variation over one slow loop sampling period

7.2.1. Brushless / DC brushed motor with quadrature encoder on motor

The internal speed units are encoder counts / (slow loop sampling period). The correspondence with the load **speed in SI units**[1] is:

For rotary motors: $$Load_Speed[SI] = \frac{2 \times \pi}{4 \times No_encoder_lines \times Tr \times T} \times Motor_Speed[IU]$$

For linear motors: $$Load_Speed[SI] = \frac{Encoder_accuracy}{Tr \times T} \times Motor_Speed[IU]$$

where:

No_encoder_lines – is the rotary encoder number of lines per revolution

Encoder_accuracy – is the linear encoder accuracy i.e. distance in [m] between 2 pulses

Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units

T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup"

7.2.2. Brushless motor with absolute SSI encoder on motor

The internal speed units are encoder counts / (slow loop sampling period). The motor is rotary. The correspondence with the load **speed in SI units**[2] is:

$$Load_Speed[SI] = \frac{2 \times \pi}{2^{No_bits_resolution} \times Tr \times T} \times Motor_Speed[IU]$$

where:

No_bits_resolution – is the SSI encoder resolution in bits per revolution

Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units

T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup"

[1] SI units for speed are [rad/s] for a rotary movement, [m/s] for a linear movement

[2] SI units for speed are [rad/s] for a rotary movement, [m/s] for a linear movement

7.2.3. DC brushed motor with quadrature encoder on load and tacho on motor

The internal speed units are encoder counts / (slow loop sampling period). The motor is rotary and the transmission is rotary-to-rotary. The correspondence with the load speed in SI units is:

$$\text{Load_Speed[SI]} = \frac{2 \times \pi}{4 \times \text{No_encoder_lines} \times T} \times \text{Load_Speed[IU]}$$

where:

No_encoder_lines – is the encoder number of lines per revolution

T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup"

7.2.4. DC brushed motor with absolute SSI encoder on load and tacho on motor

The internal speed units are encoder counts / (slow loop sampling period). The motor is rotary and the transmission is rotary-to-rotary. The correspondence with the load speed in SI units is:

$$\text{Load_Speed[SI]} = \frac{2 \times \pi}{2^{\text{No_bits_resolution}} \times T} \times \text{Load_Speed[IU]}$$

where:

No_bits_resolution – is the SSI encoder resolution in bits per revolution

T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup"

7.2.5. DC brushed motor with tacho on motor

When only a tachometer is mounted on the motor shaft, the internal speed units are A/D converter bits. The correspondence with the load **speed in SI units**[1] is:

$$\text{Load_Speed[SI]} = \frac{\text{Analogue_Input_Range}}{4096 \times \text{Tacho_gain} \times \text{Tr}} \times \text{Motor_Speed[IU]}$$

where:

Analogue_Input_Range – is the range of the drive analogue input for feedback, expressed in [V]. You can read this value in the "Drive Info" dialogue, which can be opened from the "Drive Setup"

Tacho_gain – is the tachometer gain expressed in [V/rad/s]

[1] SI units for speed are [rad/s] for a rotary movement, [m/s] for a linear movement

7.2.6. Stepper motor open-loop control. No feedback device

The internal speed units are motor µsteps / (slow loop sampling period). The correspondence with the load **speed in SI units** is:

$$\text{Load_Speed[SI]} = \frac{2 \times \pi}{\text{No_}\mu\text{steps} \times \text{No_steps} \times \text{Tr} \times \text{T}} \times \text{Motor_Speed[IU]}$$

where:

No_steps – is the number of motor steps per revolution

No_µsteps – is the number of microsteps per step. You can read/change this value in the "Drive Setup" dialogue from EasySetUp.

Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units

T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup"

7.2.7. Stepper motor open-loop control. Incremental encoder on load

The internal speed units are load encoder counts / (slow loop sampling period). The transmission is rotary-to-rotary. The correspondence with the load speed in SI units is:

$$\text{Load_Speed[rad/s]} = \frac{2 \times \pi}{4 \times \text{No_encoder_lines} \times \text{T}} \times \text{Load_Speed[IU]}$$

where:

No_encoder_lines – is the rotary encoder number of lines per revolution

Tr – transmission ratio between the motor displacement in [rad] and load displacement in [rad] or [m]

T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup".

7.2.8. Stepper motor closed-loop control. Incremental encoder on motor

The internal speed units are motor encoder counts / (slow loop sampling period). The correspondence with the load **speed in SI units**[1] is:

$$\text{Load_Speed[SI]} = \frac{2 \times \pi}{4 \times \text{No_encoder_lines} \times \text{Tr} \times \text{T}} \times \text{Motor_Speed[IU]}$$

where:

No_encoder_lines – is the motor encoder number of lines per revolution

[1] SI units for speed are [rad/s] for a rotary movement , [m/s] for a linear movement

Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units

T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup".

7.3. Acceleration units

The internal acceleration units are internal position units / (slow loop sampling period)^2 i.e. the speed variation over one slow loop sampling period.

7.3.1. Brushless / DC brushed motor with quadrature encoder on motor

The internal acceleration units are encoder counts / (slow loop sampling period)^2. The correspondence with the load **acceleration in SI units**[1] is:

For rotary motors:

$$\text{Load_Acceleration[SI]} = \frac{2 \times \pi}{4 \times \text{No_encoder_lines} \times \text{Tr} \times \text{T}^2} \times \text{Motor_Acceleration[IU]}$$

For linear motors:

$$\text{Load_Acceleration[SI]} = \frac{\text{Encoder_accuracy}}{\text{Tr} \times \text{T}^2} \times \text{Motor_Acceleration[IU]}$$

where:

No_encoder_lines – is the rotary encoder number of lines per revolution

Encoder_accuracy – is the linear encoder accuracy i.e. distance in [m] between 2 pulses

Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units

T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup"

7.3.2. Brushless motor with absolute SSI encoder on motor

The internal acceleration units are encoder counts / (slow loop sampling period)^2. The motor is rotary. The correspondence with the load acceleration in SI units is:

$$\text{Load_Acceleration[SI]} = \frac{2 \times \pi}{2^{\text{No_bits_resolution}} \times \text{Tr} \times \text{T}^2} \times \text{Motor_Acceleration[IU]}$$

[1] SI units for acceleration are [rad/s^2] for a rotary movement, [m/s^2] for a linear movement

where:

No_bits_resolution – is the SSI encoder resolution in bits per revolution

Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units

T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup"

7.3.3. DC brushed motor with quadrature encoder on load and tacho on motor

The internal acceleration units are encoder counts / (slow loop sampling period)^2. The motor is rotary and the transmission is rotary-to-rotary. The correspondence with the load acceleration in SI units is:

$$\text{Load_Acceleration[SI]} = \frac{2 \times \pi}{4 \times \text{No_encoder_lines} \times T^2} \times \text{Load_Acceleration[IU]}$$

where:

No_encoder_lines – is the encoder number of lines per revolution

T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup"

7.3.4. DC brushed motor with absolute SSI encoder on load and tacho on motor

The internal acceleration units are encoder counts / (slow loop sampling period)^2. The motor is rotary and the transmission is rotary-to-rotary. The correspondence with the load **acceleration in SI units**[1] is:

$$\text{Load_Acceleration[SI]} = \frac{2 \times \pi}{2^{\text{No_bits_resolution}} \times T^2} \times \text{Load_Acceleration[IU]}$$

where:

No_bits_resolution – is the SSI encoder resolution in bits per revolution

T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup"

[1] SI units for acceleration are [rad/s^2] for a rotary movement, [m/s^2] for a linear movement

7.3.5. DC brushed motor with tacho on motor

When only a tachometer is mounted on the motor shaft, the internal acceleration units are A/D converter bits / (slow loop sampling period). The correspondence with the load acceleration in SI units is:

$$\text{Load_Acceleration[SI]} = \frac{\text{Analogue_Input_Range}}{4096 \times \text{Tacho_gain} \times \text{Tr} \times \text{T}} \times \text{Motor_Acceleration[IU]}$$

where:

Analogue_Input_Range – is the range of the drive analogue input for feedback, expressed in [V]. You can read this value in the "Drive Info" dialogue, which can be opened from the "Drive Setup"

Tacho_gain – is the tachometer gain expressed in [V/rad/s]

T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup"

Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units

7.3.6. Stepper motor open-loop control. No feedback device

The internal acceleration units are motor μsteps / (slow loop sampling period)^2. The correspondence with the load **acceleration in SI units**[1] is:

$$\text{Load_Acceleration[SI]} = \frac{2 \times \pi}{\text{No_}\mu\text{steps} \times \text{No_steps} \times \text{Tr} \times \text{T}^2} \times \text{Motor_Acceleration[IU]}$$

where:

No_steps – is the number of motor steps per revolution

No_μsteps – is the number of microsteps per step. You can read/change this value in the "Drive Setup" dialogue from EasySetUp.

Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units

T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup"

[1] SI units for acceleration are [rad/s^2] for rotary movement, [m/s^2] for linear movement

7.3.7. Stepper motor open-loop control. Incremental encoder on load

The internal acceleration units are load encoder counts / (slow loop sampling period)^2. The correspondence with the load acceleration in SI units is:

For rotary-to-rotary transmission:

$$\text{Load_Acceleration[SI]} = \frac{2 \times \pi}{4 \times \text{No_encoder_lines} \times T^2} \times \text{Load_Acceleration[IU]}$$

For rotary-to-linear transmission:

$$\text{Load_Acceleration}[m/s^2] = \frac{\text{Encoder_accuracy}}{T^2} \times \text{Load_Acceleration[IU]}$$

where:

No_encoder_lines – is the rotary encoder number of lines per revolution

Encoder_accuracy – is the linear encoder accuracy i.e. distance in [m] between 2 pulses

Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units

T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup".

7.3.8. Stepper motor closed-loop control. Incremental encoder on motor

The internal acceleration units are motor encoder counts / (slow loop sampling period)^2. The transmission is rotary-to-rotary. The correspondence with the load **acceleration in SI units**[1] is:

$$\text{Load_Acceleration[SI]} = \frac{2 \times \pi}{4 \times \text{No_encoder_lines} \times \text{Tr} \times T^2} \times \text{Motor_Acceleration[IU]}$$

where:

No_encoder_lines – is the motor encoder number of lines per revolution

Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units

T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup"

[1] SI units for acceleration are [rad/s^2] for rotary movement, [m/s^2] for linear movement

7.4. Jerk units

The internal jerk units are internal position units / (slow loop sampling period)^3 i.e. the acceleration variation over one slow loop sampling period.

7.4.1. Brushless / DC brushed motor with quadrature encoder on motor

The internal jerk units are encoder counts / (slow loop sampling period)^3. The correspondence with the load **jerk in SI units**[1] is:

For rotary motors: $$Load_Jerk[SI] = \frac{2 \times \pi}{4 \times No_encoder_lines \times Tr \times T^3} \times Motor_Jerk[IU]$$

For linear motors: $$Load_Jerk[SI] = \frac{Encoder_accuracy}{Tr \times T^3} \times Motor_Jerk[IU]$$

where:

No_encoder_lines – is the rotary encoder number of lines per revolution

Encoder_accuracy – is the linear encoder accuracy i.e. distance in [m] between 2 pulses

Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units

T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup"

7.4.2. Brushless motor with absolute SSI encoder on motor

The internal jerk units are encoder counts / (slow loop sampling period)^3. The motor is rotary. The correspondence with the load **jerk in SI units**[2] is:

$$Load_Jerk[SI] = \frac{2 \times \pi}{2^{No_bits_resolution} \times Tr \times T^3} \times Motor_Jerk[IU]$$

where:

No_bits_resolution – is the SSI encoder resolution in bits per revolution

Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units

T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialog, which can be opened from the "Drive Setup"

[1] SI units for jerk are [rad/s^3] for a rotary movement, [m/s^3] for a linear movement

[2] SI units for jerk are [rad/s^3] for a rotary movement, [m/s^3] for a linear movement

7.4.3. DC brushed motor with quadrature encoder on load and tacho on motor

The internal jerk units are encoder counts / (slow loop sampling period)^3. The motor is rotary and the transmission is rotary-to-rotary. The correspondence with the load jerk in SI units is:

$$\text{Load_Jerk[SI]} = \frac{2 \times \pi}{4 \times \text{No_encoder_lines} \times T^3} \times \text{Load_Jerk[IU]}$$

where:

No_encoder_lines – is the encoder number of lines per revolution

T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialog, which can be opened from the "Drive Setup"

7.4.4. DC brushed motor with absolute SSI encoder on load and tacho on motor

The internal jerk units are encoder counts / (slow loop sampling period)^3. The motor is rotary and the transmission is rotary-to-rotary. The correspondence with the load jerk in SI units is:

$$\text{Load_Jerk[SI]} = \frac{2 \times \pi}{2^{\text{No_bits_resolution}} \times T^2} \times \text{Load_Jerk[IU]}$$

where:

No_bits_resolution – is the SSI encoder resolution in bits per revolution

T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialog, which can be opened from the "Drive Setup"

7.4.5. Stepper motor open-loop control. No feedback device

The internal jerk units are motor µsteps / (slow loop sampling period)^3. The correspondence with the load **jerk in SI units**[1] is:

$$\text{Load_Jerk[SI]} = \frac{2 \times \pi}{\text{No_}\mu\text{steps} \times \text{No_steps} \times \text{Tr} \times T^3} \times \text{Motor_Jerk[IU]}$$

where:

No_steps – is the number of motor steps per revolution

No_µsteps – is the number of microsteps per step. You can read/change this value in the "Drive Setup" dialog from EasySetUp.

Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units

T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialog, which can be opened from the "Drive Setup"

[1] SI units for jerk are [rad/s^3] for a rotary movement, [m/s^3] for a linear movement

7.4.6. Stepper motor open-loop control. Incremental encoder on load

The internal jerk units are load encoder counts / (slow loop sampling period)^3. The transmission is rotary-to-rotary. The correspondence with the load jerk in SI units is:

$$\text{Load_Jerk[SI]} = \frac{2 \times \pi}{4 \times \text{No_encoder_lines} \times T^3} \times \text{Load_Jerk[IU]}$$

where:

No_encoder_lines – is the rotary encoder number of lines per revolution

T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialog, which can be opened from the "Drive Setup".

7.4.7. Stepper motor closed-loop control. Incremental encoder on motor

The internal jerk units are motor encoder counts / (slow loop sampling period)^3. The correspondence with the load jerk in SI units is:

$$\text{Load_Jerk[SI]} = \frac{2 \times \pi}{4 \times \text{No_encoder_lines} \times Tr \times T^3} \times \text{Motor_Jerk[IU]}$$

where:

No_encoder_lines – is the motor encoder number of lines per revolution

Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units

T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialog, which can be opened from the "Drive Setup".

7.5. Current units

The internal current units refer to the motor phase currents. The correspondence with the motor currents in [A] is:

$$\text{Current[A]} = \frac{2 \times \text{Ipeak}}{65520} \times \text{Current[IU]}$$

where Ipeak – is the drive peak current expressed in [A]. You can read this value in the "Drive Info" dialogue, which can be opened from the "Drive Setup".

7.6. Voltage command units

The internal voltage command units refer to the voltages applied on the motor. The significance of the voltage commands as well as the scaling factors, depend on the motor type and control method used.

In case of **brushless motors** driven in **sinusoidal** mode, a field oriented vector control is performed. The voltage command is the amplitude of the sinusoidal phase voltages. In this case, the correspondence with the motor phase voltages in SI units i.e. [V] is:

$$\text{Voltage command[V]} = \frac{1.1 \times \text{Vdc}}{65534} \times \text{Voltage command[IU]}$$

where Vdc – is the drive power supply voltage expressed in [V].

In case of **brushless** motors driven in **trapezoidal** mode, the voltage command is the voltage to apply between 2 of the motor phases, according with Hall signals values. In this case, the correspondence with the voltage applied in SI units i.e. [V] is:

$$\text{Voltage command[V]} = \frac{\text{Vdc}}{32767} \times \text{Voltage command[IU]}$$

This correspondence is also available for **DC brushed** motors which have the voltage command internal units as the brushless motors driven in trapezoidal mode.

7.7. Voltage measurement units

The internal voltage measurement units refer to the drive V_{MOT} supply voltage. The correspondence with the supply voltage in [V] is:

$$\text{Voltage_measured[V]} = \frac{\text{VdcMaxMeasurable}}{65520} \times \text{Voltage_measured[IU]}$$

where VdcMaxMeasurable – is the maximum measurable DC voltage expressed in [V]. You can read this value in the “Drive Info” dialogue, which can be opened from the “Drive Setup”.

Remark: *The voltage measurement units occur in the scaling of the over voltage and under voltage protections and the supply voltage measurement*

7.8. Time units

The internal time units are expressed in slow loop sampling periods. The correspondence with the time in [s] is:

$$\text{Time[s]} = T \times \text{Time[IU]}$$

where T – is the slow loop sampling period expressed in [s]. You can read this value in the “Advanced” dialogue, which can be opened from the “Drive Setup”. For example, if T = 1ms, one second = 1000 IU.

7.9. Drive temperature units

The motion controller includes a temperature sensor. The correspondence with the temperature in [°C] is:

$$\text{Drive temperature [°C]} = \frac{3[\text{V}] \times \text{DriveTemperature[IU]}}{65520 \times \text{Sensor_gain[V/°C]}} - \frac{\text{Sensor_output_0°C[V]}}{\text{Sensor_gain[V/°C]}}$$

where:

Sensor_gain – is the temperature sensor gain

Sensor_output_0°C – is the temperature sensor output at 0°C. You can read these values in the "Drive Info" dialogue, which can be opened from the "Drive Setup"

7.10. Motor position units

7.10.1. Brushless / DC brushed motor with quadrature encoder on motor

The internal motor position units are encoder counts. The correspondence with the motor **position in SI units**[1] is:

For rotary motors: $$\text{Motor_Position[SI]} = \frac{2 \times \pi}{4 \times \text{No_encoder_lines}} \times \text{Motor_Position[IU]}$$

For linear motors: $$\text{Motor_Position[SI]} = \text{Encoder_accuracy} \times \text{Motor_Position[IU]}$$

where:

No_encoder_lines – is the rotary encoder number of lines per revolution

Encoder_accuracy – is the linear encoder accuracy i.e. distance in [m] between 2 pulses

7.10.2. Brushless motor with absolute SSI encoder on motor

The internal motor position units are encoder counts. The motor is rotary. The correspondence with the motor **position in SI units**[2] is:

$$\text{Motor_Position[SI]} = \frac{2 \times \pi}{2^{\text{No_bits_resolution}}} \times \text{Motor_Position[IU]}$$

where:

No_bits_resolution – is the SSI encoder resolution in bits per revolution

[1] SI units for motor position are: [rad] for a rotary motor, [m] for a linear motor

[2] SI units for motor position are: [rad] for a rotary motor, [m] for a linear motor

7.10.3. DC brushed motor with quadrature encoder on load and tacho on motor

The motor position is not computed.

7.10.4. DC brushed motor with absolute SSI encoder on load and tacho on motor

The motor position is not computed.

7.10.5. Stepper motor open-loop control. No feedback device

The internal motor position units are motor μsteps. The correspondence with the motor **position in SI units**[1] is:

$$\text{Motor_Position[SI]} = \frac{2 \times \pi}{\text{No_}\mu\text{steps} \times \text{No_steps}} \times \text{Motor_Position[IU]}$$

where:

No_steps – is the number of motor steps per revolution

No_μsteps – is the number of microsteps per step. You can read/change this value in the "Drive Setup" dialogue from EasySetUp.

7.10.6. Stepper motor open-loop control. Incremental encoder on load

In open-loop control configurations with incremental encoder on load, the motor position is not computed.

7.10.7. Stepper motor closed-loop control. Incremental encoder on motor

The internal motor position units are motor encoder counts. The correspondence with the motor position in SI units is:

$$\text{Motor_Position[SI]} = \frac{2 \times \pi}{4 \times \text{No_encoder_lines}} \times \text{Motor_Position[IU]}$$

where:

No_encoder_lines – is the motor encoder number of lines per revolution

[1] SI units for motor position are [rad] for a rotary motor, [m] for a linear motor

7.11. Motor speed units

7.11.1. Brushless / DC brushed motor with quadrature encoder on motor

The internal motor speed units are encoder counts / (slow loop sampling period). The correspondence with the motor **speed in SI units**[1] is:

For rotary motors: $$Motor_Speed[SI] = \frac{2 \times \pi}{4 \times No_encoder_lines \times T} \times Motor_Speed[IU]$$

For linear motors: $$Motor_Speed[SI] = \frac{Encoder_accuracy}{T} \times Motor_Speed[IU]$$

where:

No_encoder_lines – is the rotary encoder number of lines per revolution

Encoder_accuracy – is the linear encoder accuracy i.e. distance in [m] between 2 pulses

T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup"

7.11.2. Brushless motor with absolute SSI encoder on motor

The internal motor speed units are encoder counts / (slow loop sampling period). The motor is rotary. The correspondence with the motor **speed in SI units**[2] is:

$$Motor_Speed[SI] = \frac{2 \times \pi}{2^{No_bits_resolution} \times T} \times Motor_Speed[IU]$$

where:

No_bits_resolution – is the SSI/BiSS encoder resolution in bits per revolution

T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup"

[1] SI units for motor speed are [rad/s] for a rotary motor, [m/s] for a linear motor

[2] SI units for motor speed are [rad/s] for a rotary motor, [m/s] for a linear motor

7.11.3. DC brushed motor with quadrature encoder on load and tacho on motor

The internal motor speed units are A/D converter bits. The correspondence with the motor **speed in SI units**[1] is:

$$\text{Motor_Speed[SI]} = \frac{\text{Analogue_Input_Range}}{4096 \times \text{Tacho_gain}} \times \text{Motor_Speed[IU]}$$

where:

Analogue_Input_Range – is the range of the drive analogue input for feedback, expressed in [V]. You can read this value in the “Drive Info” dialogue, which can be opened from the “Drive Setup”

Tacho_gain – is the tachometer gain expressed in [V/rad/s]

7.11.4. DC brushed motor with absolute SSI encoder on load & tacho on motor

The internal motor speed units are A/D converter bits. The correspondence with the motor speed in SI units is:

$$\text{Motor_Speed[SI]} = \frac{\text{Analogue_Input_Range}}{4096 \times \text{Tacho_gain}} \times \text{Motor_Speed[IU]}$$

where:

Analogue_Input_Range – is the range of the drive analogue input for feedback, expressed in [V]. You can read this value in the “Drive Info” dialogue, which can be opened from the “Drive Setup”

Tacho_gain – is the tachometer gain expressed in [V/rad/s]

7.11.5. DC brushed motor with tacho on motor

The internal motor speed units are A/D converter bits. The correspondence with the motor speed in SI units is:

$$\text{Motor_Speed[SI]} = \frac{\text{Analogue_Input_Range}}{4096 \times \text{Tacho_gain}} \times \text{Motor_Speed[IU]}$$

where:

Analogue_Input_Range – is the range of the drive analogue input for feedback, expressed in [V]. You can read this value in the “Drive Info” dialogue, which can be opened from the “Drive Setup”

Tacho_gain – is the tachometer gain expressed in [V/rad/s]

[1] SI units for motor speed are [rad/s] for a rotary motor, [m/s] for a linear motor

7.11.6. Stepper motor open-loop control. No feedback device or incremental encoder on load

The internal motor speed units are motor μsteps / (slow loop sampling period). The correspondence with the motor **speed in SI units**[1] is:

$$\text{Motor_Speed[SI]} = \frac{2 \times \pi}{\text{No_}\mu\text{steps} \times \text{No_steps} \times T} \times \text{Motor_Speed[IU]}$$

where:

No_steps – is the number of motor steps per revolution

No_μsteps – is the number of microsteps per step. You can read/change this value in the "Drive Setup" dialogue from EasySetUp.

T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup"

7.11.7. Stepper motor closed-loop control. Incremental encoder on motor

The internal motor speed units are motor encoder counts / (slow loop sampling period). The correspondence with the load speed in SI units is:

$$\text{Motor_Speed[SI]} = \frac{2 \times \pi}{4 \times \text{No_encoder_lines} \times T} \times \text{Motor_Speed[IU]}$$

where:

No_encoder_lines – is the motor encoder number of lines per revolution

T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup".

[1] SI units for motor speed are [rad/s] for a rotary motor, [m/s] for a linear motor

This page is empty

8. Memory Map

TMC-3D has 2 types of memory available for user applications: 4K×16 SRAM and 8K×16 serial E^2ROM.

The SRAM memory is mapped in the address range: 9000h to 9FFFh. It can be used to download and run a TML program, to save real-time data acquisitions and to keep the cam tables during run-time.

The E^2ROM is mapped in the address range: 4000h to 5FFFh. It is used to keep in a non-volatile memory the TML programs, the cam tables and the drive setup information.

Remark: *EasyMotion Studio handles automatically the memory allocation for each motion application. The memory map can be accessed and modified from the main folder of each application*

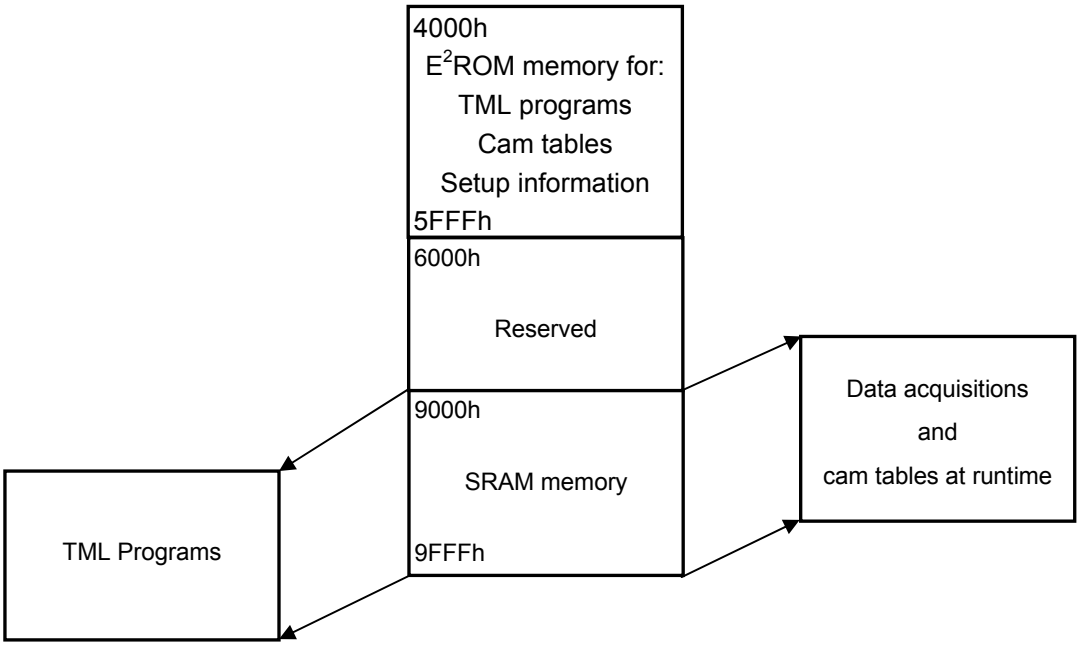


Figure 8.1. *TMC-3D Memory Map*

This page is empty

TECHNOSOFT